

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»  
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ  
*Кафедра автоматизованих систем обробки інформації і управління*

До захисту допущено:

В.о. завідувача кафедри

\_\_\_\_\_  
(підпис) Олександр ПАВЛОВ  
(вл.ім'я, прізвище)

“ \_\_\_\_ ” \_\_\_\_\_ 2020 р.

**Дипломний проєкт**  
**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою «Інформаційні управляючі  
системи та технології»  
спеціальності 122 «Комп'ютерні науки та інформаційні технології»**

на тему: *«Система аналізу та прогнозування інформаційно-  
комунікаційних процесів»*

**Виконав:**

студент IV курсу, групи ІС-63

\_\_\_\_\_  
*Процюк Юрій Володимирович*  
(прізвище, ім'я, по батькові)

\_\_\_\_\_  
(підпис)

**Керівник**

\_\_\_\_\_  
*доц., к.т.н., доц. Жаріков Едуард В'ячеславович*  
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

\_\_\_\_\_  
(підпис)

**Консультант з  
графічної  
документації**

\_\_\_\_\_  
*ст. викладач Проскура Світлана Леонідівна*  
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

\_\_\_\_\_  
(підпис)

**Рецензент**

\_\_\_\_\_  
*доц. каф. ТК, к.т.н., доц. Лісовиченко Олег Іванович*  
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

\_\_\_\_\_  
(підпис)

Засвідчую, що у цьому дипломному проєкті  
немає запозичень з праць інших авторів без  
відповідних посилань.

Студент (-ка) \_\_\_\_\_  
(підпис)

Київ – 2020 року

**Національний технічний університет України**  
**“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) інформатики та обчислювальної техніки  
(повна назва)

Кафедра автоматизованих систем обробки інформації і управління  
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки та інформаційні технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

Олександр ПАВЛОВ  
(підпис) (вл.ім'я, прізвище)

“ ” 2020 р.

**ЗАВДАННЯ**  
**на дипломний проєкт студенту**

Процюку Юрію Володимировичу  
(прізвище, ім'я, по батькові)

1. Тема проєкту «Система аналізу та прогнозування інформаційно-комунікаційних процесів»

керівник проєкту Жаріков Едуард В'ячеславович, к.т.н., доцент  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “7” травня 2020 р. №1081-с

2. Термін подання студентом проєкту “01” червня 2020 року

3. Вихідні дані до проєкту

Технічне завдання

4. Зміст пояснювальної записки

1. Загальні положення: основні визначення та терміни, опис предметного середовища, огляд ринку програмних продуктів, постановка задачі

2. Інформаційне забезпечення: вхідні дані, вихідні дані, опис структури бази даних

3. Математичне забезпечення: змістовна та математична постановки задачі, обґрунтування та опис методу розв'язання

4. Програмне та технічне забезпечення: засоби розробки, вимоги до технічного забезпечення, архітектура програмного забезпечення, побудова звітів

5. Технологічний розділ: керівництво користувача, методика випробувань програмного продукту

5. Перелік графічного матеріалу

1. Схема структурна діяльності

1. Схема структурна варіантів використання

2. Схема бази даних

3. Схема структурна компонентів програмного забезпечення

4. Схема структурна класів програмного забезпечення

6. Схема структурна послідовності

6. Креслення вигляду екранних форм

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «13» квітня 2020 року

### Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	Вивчення рекомендованої літератури	15.03.2020	
2.	Аналіз існуючих методів розв'язання задачі	25.03.2020	
3.	Постановка та формалізація задачі	10.04.2020	
4.	Розробка інформаційного забезпечення	15.04.2020	
5.	Алгоритмізація задачі	20.04.2020	
6.	Обґрунтування використовуваних технічних засобів	25.05.2020	
7.	Розробка програмного забезпечення	01.05.2020	
8.	Налагодження програми	05.05.2020	
9.	Виконання графічних документів	15.05.2020	
10.	Оформлення пояснювальної записки	20.05.2020	
11.	Подання ДП на попередній захист	31.05.2020	
12.	Подання ДП на основний захист	01.06.2020	
13.	Подання ДП рецензенту	02.06.2020	

Студент

Юрій ПРОЦЮК

Керівник

Едуард ЖАРИКОВ

[illegible]

# **Пояснювальна записка до дипломного проєкту**

на тему: Система аналізу та прогнозування інформаційно-комунікаційних процесів

---

Київ – 2020 року

## АНОТАЦІЯ

**Структура та обсяг роботи.** Пояснювальна записка дипломного проєкту складається з п'яти розділів, містить 20 рисунків, 29 таблиць, 1 додаток, 19 джерел.

Дипломний проєкт присвячений розробленню системи аналізу та прогнозування інформаційно-комунікаційних процесів. Метою дипломного проєкту є покращення організації діяльності з аналізу та прогнозування інформаційно-комунікаційних процесів. Для досягнення поставленої мети необхідно розв'язати такі задачі: провести аналіз процесів, з визначенням параметрів, що найбільше впливають на процеси; на основі проведеного аналізу процесів, визначити методи для прогнозування стану виконання процесів; розробити програмний застосунок, що здійснює прогнозування плинності процесів; підтримка можливості збереження результату прогнозування; провести тестування й оцінку продуктивності розробленого застосунку.

Розділ загальних положень описує предметне середовище, процес діяльності та функціональну модель системи. Порівняно систему з аналогами, вказано подібності та відмінності.

У розділі інформаційного забезпечення визначено вхідні та вихідні дані до системи, надано їх детальний опис з поясненням їх призначення.

Розділ математичного забезпечення присвячений формулюванню та розв'язку формалізованих проблем, що вирішуються системою. Надано обґрунтування обраних методів розв'язання та їх детальний опис.

Розділ програмного забезпечення надає опис засобів та технологій, що застосовувалися при розробці, також формулює вимоги до технічного

					ДП 6319.00.000 ПЗ								
		Прізвище	Підпис	Дата									
Розроб.	Процюк Ю.В..				Система аналізу та прогнозування інформаційно- комунікаційних процесів			Літ.		Лист		Листів	
Перевірив.	Жаріков Е.В.									2		85	
								КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-63					
Н. кон.	Проскура С.Л.												
Затв.	Павлов О.А.												

забезпечення. У розділі описано використаний архітектурний шаблон.  
Описано семантику деяких важливих методів програми.

У технологічному розділі надано детальну з рисунками користувацьку інструкцію та проведено тестування системи на відповідність вимогам.

АНАЛІЗ ПРОЦЕСІВ, ПРОГНОЗУВАННЯ ПРОЦЕСІВ, НЕЙРОННІ МЕРЕЖІ, МЕТОД КРИТИЧНОГО ШЛЯХУ

					ДП 6319.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

## ABSTRACT

**Structure and scope of work.** The explanatory note of the diploma project consists of five sections, contains 20 figures, 29 tables, 1 appendix, 19 sources.

The diploma project is devoted to the development of a system of analysis and forecasting of information and communication processes. The purpose of the diploma project is to improve the organization of activities for the analysis and forecasting of information and communication processes. To achieve this goal it is necessary to solve the following tasks: to analyze the processes, identifying the parameters that most affect the processes; on the basis of the conducted analysis of processes, to define methods for forecasting of a condition of performance of processes; develop a software application that predicts the flow of processes; support for the ability to save the forecast result; to test and evaluate the performance of the developed application.

The section of general provisions describes the subject environment, the process of activity and the functional model of the system. The system is compared with analogues, similarities and differences are indicated.

In the section of information support the input and output data to the system are defined, their detailed description with the explanation of their purpose is given.

The section of mathematical support is devoted to the formulation and solution of formalized problems solved by the system. The substantiation of the chosen methods of the decision and their detailed description is given.

The software section provides a description of the tools and technologies used in the development, and also formulates the hardware requirements. The section describes the architectural template used. The semantics of some important methods of the program are described.

The technological section provides detailed user instructions with drawings and tests of the system for compliance with the requirements.

					ДП 6319.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4



PROCESS ANALYSIS, PROCESS FORECASTING, NEURAL NETWORKS, CRITICAL PATH METHOD

					ДП 6319.00.000 ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

## ЗМІСТ

ВСТУП .....	8
<b>1 ЗАГАЛЬНІ ПОЛОЖЕННЯ .....</b>	<b>11</b>
1.1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА .....	11
1.1.1 Опис процесу діяльності .....	12
1.1.2 Опис функціональної моделі .....	13
1.2 ОГЛЯД НАЯВНИХ АНАЛОГІВ .....	14
1.3 ПОСТАНОВКА ЗАДАЧІ .....	15
1.3.1 Призначення розробки .....	15
1.3.2 Цілі та задачі розробки .....	15
Висновок до розділу .....	16
<b>2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ .....</b>	<b>17</b>
2.1 ВХІДНІ ДАНІ .....	17
2.2 ВИХІДНІ ДАНІ .....	18
2.3 ОПИС СТРУКТУРИ БАЗИ ДАНИХ .....	18
Висновок до розділу .....	24
<b>3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ .....</b>	<b>25</b>
3.1 ЗМІСТОВНА ПОСТАНОВКА ЗАДАЧІ .....	25
3.1.1 Змістовна постановка задачі 1 .....	25
3.1.2 Змістовна постановка задачі 2 .....	25
3.2 ФОРМАЛІЗОВАНА ПОСТАНОВКА ЗАДАЧІ .....	25
3.2.1 Формалізована постановка задачі 1 .....	25
3.2.2 Формалізована постановка задачі 2 .....	25
3.3 ОБҐРУНТУВАННЯ МЕТОДІВ РОЗВ'ЯЗАННЯ .....	26
3.4 ОПИС МЕТОДІВ РОЗВ'ЯЗАННЯ .....	26
3.4.1 Опис методу розв'язання задачі 1 .....	26
3.4.2 Опис методу розв'язання задачі 2 .....	29
Висновок до розділу .....	32
<b>4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ .....</b>	<b>33</b>
4.1 ЗАСОБИ РОЗРОБКИ .....	33
4.2 ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ .....	39

<b>4.3</b>	<b>АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....</b>	<b>39</b>
4.3.1	Діаграма класів .....	41
4.3.2	Діаграма послідовності.....	41
4.3.3	Діаграма компонентів.....	42
4.3.4	Специфікація функцій .....	42
	Висновок до розділу .....	48
<b>5</b>	<b>ТЕХНОЛОГІЧНИЙ РОЗДІЛ .....</b>	<b>50</b>
5.1	КЕРІВНИЦТВО КОРИСТУВАЧА .....	50
5.2	ВИПРОБУВАННЯ ПРОГРАМНОГО ПРОДУКТУ .....	59
5.2.1	Мета випробувань .....	59
5.2.2	Загальні положення .....	59
5.2.3	Результати випробувань .....	60
	Висновок до розділу .....	68
	ЗАГАЛЬНІ ВИСНОВКИ .....	69
	ПЕРЕЛІК ПОСИЛАНЬ .....	71
	ДОДАТОК А .....	73

## ВСТУП

У сучасному світі людині постійно доводиться стикатися із процесами. У школі та інших навчальних закладах постійно відбувається освітній процес, батьки, школа та суспільство в цілому допомагають людині пройти процес виховання. Загальнонауковими є процеси апробації чи впровадження. На усіх підприємствах є специфічні процеси. В залежності від виду діяльності компанії мають свої особливі процеси, наприклад телекомунікаційні компанії – процеси виставлення рахунків, логістичні - транспортні процеси, аудиторські фірми – процеси перевірки. Існує величезна кількість визначень даного поняття, але чи не найбільш містким та зрозумілим є таке: процес – це зазвичай фіксована чи впорядкована послідовність дій чи подій, що призводять до результату [1]. З цього визначення випливають такі важливі ознаки процесу, як відсутність атомарності та наявність наслідку виконання процесу. Ці властивості свідчать про те, що будь-який процес має свою важливість. Складними та важливими водночас є інформаційно-комунікаційні процеси, тобто ті процеси, що передбачають взаємодію кількох елементів, у тому числі на рівні обміну інформацією. Інформаційно-комунікаційні процеси відбуваються повсюди, зокрема і в суб'єктах господарювання – підприємствах.

Розглядати конкретні процеси, що відбуваються на підприємствах, то вони теж дуже важливі, бо достеменно відомо, що результати процесів будуть застосовуватися у майбутньому. Поміж іншого може бути відомо навіть точну сферу застосування результатів процесу. Тому постає питання про те, щоб процеси не просто виконувалися, а виконувались таким чином, щоб бути максимально корисними та найкраще організованими у структурі діяльності підприємства. Для цього потрібно визначити те, як саме мають бути виконані процеси – які їх аспекти найбільше впливають на загальний результат, а які, навпаки, майже не несуть навантаження чи не мають цінності. Серед варіантів визначення основних параметрів роботи процесу

					ДП 6319.00.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

можна назвати упорядкування за важливістю, вибір елементів за певним критерієм. Але ці способи мають суттєвий недолік – вони не точні та не універсальні. Стандарт ISO 9000:2008 пропонує метод визначення ключових показників ефективності (KPI – Key Performance Indicators).

Ключові показники ефективності – це показники, які зосереджені на аспектах діяльності організації, які є найважливішими для поточного та майбутнього успіху організації [2]. Загальний підхід до формулювання ключових показників ефективності такий:

- а) визначити стратегічні цілі;
- б) визначити результат, що буде успішним;
- в) сформулювати показники ефективності за принципом smart (конкретні, вимірювані, досяжні, доцільні та обмежені у часі).

Сформульовані показники повинні враховувати такі властивості процесу: дієвість, ефективність, час проходження, якість результатів та відповідність нормативним вимогам. Дієвість в контексті процесу означає зв'язок між фактичними та очікуваними результатами. Ефективність є зв'язком між результатами, досягнутими в процесі, та ресурсами, що були затратені та спожиті. Відповідність процесу поділяється на внутрішню та зовнішню. Внутрішня визначає чи досягнуто в результаті виконання процесу бажаного результату (або ж частка досягнутого результату). Зовнішня відповідність – контролює узгодженість зі стандартами та наявними нормами. Час проходження – визначає часові витрати на виконання процесу, враховуючи всі етапи та підпроцеси і їх зв'язки. Якість результатів процесу – чи отримані результати відповідають очікуванням їх споживача, внутрішнім вимогам, в тому числі й бюджетним. Саме ці аспекти й визначають статус процесу, допомагають в прогнозі його результатів і можливих відхилень [3].

Дипломний проєкт присвячений розробленню системи аналізу та прогнозування інформаційно-комунікаційних процесів.

					ДП 6319.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

**Практичне значення одержаних результатів.** Запропоновано підхід до прогнозування стану плинущих процесів.

**Публікації.** Результати роботи були опубліковані у тезах доповідей IV Всеукраїнської науково-практичної конференції молодих вчених та студентів «Інформаційні системи та технології управління – ІСТУ-2020». Секція кафедри автоматизованих систем обробки інформації та управління [3].

					ДП 6319.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

# 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

## 1.1 Опис предметного середовища

Уся діяльність підприємств та різноманітних організацій зосереджена на процесах. Без процесів не можливо уявити діяльність жодної компанії. Процеси можуть бути основними, управлінськими, процесами забезпечення та розвитку. За ступенем автоматизації вони можуть бути ручними, автоматизованими та автоматичними (від нульового ступеня автоматизації до повної відсутності втручання людини). Процеси в штучних системах, що створені людиною, можна назвати інформаційно-комунікаційними, якщо їх діяльність передбачає обмін інформацією [4].

Процес – складна сутність, неможливо точно оцінити його стан через певний відтинок часу, оскільки на плин процесу впливає величезна кількість різноманітних факторів. Це і є основною складністю в задачі аналізу та прогнозування плину процесів. Наприклад, процес підключення домашнього інтернету (автоматизований, бо має в собі й автоматичні аспекти й ручне налаштування), коли його буде завершено і можна буде користуватися інтернетом. Це залежить від багатьох факторів. Основними є дата початку та очікувана тривалість, оцінка якої може спиратися на попередній досвід. Також параметрами, що впливають на час завершення підключення, є досвід спеціаліста, що проводить підключення, фізична складність доступу до місця підключення, необхідність залучення додаткових компонент чи елементів до виконання процесу та інші фактори. Аналіз усіх параметрів дозволяє оцінити дату завершення підключення, що своєю чергою дозволить прогнозувати виконання інших процесів, залучати співробітників до інших задач та приймати велику кількість важливих управлінських рішень.

Важливим аспектом аналізу процесів є визначення ланцюгів процесів. Кожен процес може бути елементом ланцюга процесів, тобто знаходитися в залежності від інших процесів. Визначення того, який процесам передую

					ДП 6319.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

даний (у тому числі пошук найдовшого ланцюга), може дозволити прийняти рішення про залучення додаткових ресурсів на виконання процесу, призначити більш досвідченого виконавця, чи можливо декількох виконавців. Ці аналітичні дані дозволять також перерозподіляти ресурси на виконання інших процесів, що не залежать від даного.

Для того, щоб провести адекватне прогнозування процесів, важливо визначити те, які параметри найбільше впливають на плин процесу та яким саме є цей вплив.

### 1.1.1 Опис процесу діяльності

Об'єктом автоматизації є діяльність з аналізу та прогнозування інформаційно-комунікаційних процесів, точніше кажучи пошук найдовшого ланцюга процесів, що починається з обраного процесу і прогнозування часу завершення плину процесу.

Щоб правильно виконати проєктування потрібно визначити акторів застосунку. Акторами є аналітик та аналітик з правами адміністратора. Вважатимемо цих акторів користувачами системи.

Процес використання застосунку для аналітика можна зобразити за допомогою діаграми діяльності (розміщена в графічному матеріалі).

Користувач при вході в систему повинен пройти процедуру авторизації (для уже наявних користувачів) чи реєстрації (для нових користувачів). Далі він повинен перейти на головну сторінку, де подано інформацію про процеси. На цій сторінці він може обрати такі варіанти:

- «Прогнозування плину процесів», що виконає прогнозування часу завершення процесу і час проміжного виконання;
- «Пошук ланцюга», де користувач перегляне найдовший ланцюг, що починається з даного процесу.

Додатковими можливостями користувача із правами адміністратора є змога редагувати інших користувачів, додавати нові компоненти.

					ДП 6319.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12



### 1.1.2 Опис функціональної моделі

Особливості функціональної поведінки системи представлено у вигляді варіантів використання. Структурну схему варіантів використання наведено у графічному матеріалі.

Акторами системи є аналітик та аналітик-адміністратор, вони ж і є користувачами застосунку.

Дії чи варіанти виконання, що можуть бути виконані акторами, наведені в таблиці 1.1, у якій наявний опис актора, варіантів використання та дія, що є в основі варіанту використання.

Таблиця 1.1 – Варіанти використання та їх опис

Актор	Варіант використання	Опис варіанту використання
Аналітик	Реєстрація/авторизація	Система надає користувачеві змогу увійти в систему/ зареєструватися у системі
	Додання процесу	Система надає користувачеві змогу створити новий процес і додати його в систему.
	Пошук найдовшого ланцюга	Система надає користувачеві змогу обрати процес, який буде початковим для пошуку найдовшого ланцюга.
	Прогнозування часу завершення процесу	Система надає користувачеві змогу обрати кількість процесів, на яких відбуватиметься навчання, для прогнозування часу завершення інших процесів.

## Продовження таблиці 1.1

Актор	Варіант використання	Опис варіанту використання
Аналітик-адміністратор	Додання компонентів	Система надає користувачеві з правами адміністратора змогу додавати компоненти.
	Редагування користувачів	Система надає користувачеві з правами адміністратора змогу виконувати редагування інших користувачів.

**1.2 Огляд наявних аналогів**

У результаті огляду наявних аналогів зроблено висновок, що абсолютно аналогічних застосунків немає. Це обумовлено насамперед тим, що в кожній компанії процеси досить специфічні й нема застосунків для аналізу та прогнозування процесів у відкритому доступі. Існують системи та продукти для загального аналізу даних та їх прогнозування, але вони не орієнтовані саме на процеси.

Не можна оминути, зокрема, дуже потужний пакет обробки даних «Statistica», що має широкий набір функцій, зокрема основні статистики та графічні інструменти для аналізу даних. Також в розширенні системи «Statistica» «Statistica Process Analysis» доступні можливості аналізу придатності процесу, аналіз повторюваності та відтворюваності вимірів. Ще «Statistica» надає широкі можливості побудови нейронних мереж для різних задач, але цей пакет не є найкращим для саме аналізу та прогнозування інформаційно-комунікаційних процесів через їх специфічні характеристики та властивості. Загалом «Statistica» – дуже потужний інструмент, основним недоліком якого і є універсальність, а не орієнтування на певну специфічну

задачу. Серед переваг «Statistica» – російськомовний інтерфейс користувача, що є зручним для тих, хто не знає англійської мови.

Мова програмування R, що є водночас і середовищем аналізу даних, теж може розглядатися як аналог для розроблюваного застосунку, але не є такою через те, що не оперує поняттям аналізу та прогнозування процесу. Серед переваг R – широке використання і фактичне використання як стандарту для статистичних програм. Перевагою R є і наявність значної кількості графічних інтерфейсів для зручності роботи. Серед недоліків мови R ще відзначають те, що вона не може бути вбудована у вебпереглядач, тобто не можна її використовувати для розробки веб застосунків [5].

### 1.3 Постановка задачі

#### 1.3.1 Призначення розробки

Призначенням системи є аналіз процесів та прогнозування стану їх виконання.

#### 1.3.2 Цілі та задачі розробки

Метою дипломного проєкту є покращення організації діяльності з аналізу та прогнозування інформаційно-комунікаційних процесів.

Для досягнення поставленої мети необхідно розв'язати такі задачі:

- провести аналіз процесів, з визначенням параметрів, що найбільше впливають на процеси;
- на основі проведеного аналізу процесів, визначення методів для прогнозування стану виконання процесів;
- розробити програмний застосунок, що здійснює прогнозування плинну процесів;
- підтримка можливості збереження результату прогнозування;
- провести тестування й оцінку продуктивності розробленого застосунку.

					ДП 6319.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

Система може застосовуватись як і окремо, так і вбудовуватися в іншу систему.

### Висновок до розділу

У розділі «Загальні положення» було розглянуто предметне середовище, зокрема визначено особливості аналізу та прогнозування інформаційно-комунікаційних процесів, зважаючи на характеристики процесів.

При описі предметного середовища було виконано опис процесів діяльності системи, визначено їх особливості. Побудовано функціональну модель системи, яка представлена у вигляді діаграми варіантів використання. В use-case діаграмі визначено акторів, що діють у системі, а також надано функціональні вимоги, що відповідають варіантам використання.

Також виконано огляд аналогів запропонованого застосунку, що мають схожі функції. Проведено порівняльний аналіз окремих наявних рішень для аналізу та прогнозування інформаційно-комунікаційних процесів та визначено, що вони не є строго орієнтовані на процеси, а є системами загального аналізу та прогнозування даних, зокрема статистичного аналізу. У розділі сформовано постановку задачі, призначення та мету розробки застосунку. Дипломний проєкт присвячено розробці системи аналізу та прогнозування інформаційно-комунікаційних процесів.

## 2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

### 2.1 Вхідні дані

Вхідними даними для розроблюваної системи є реальні (анонімізовані) дані про інформаційно-комунікаційні процеси:

- а) процеси для аналізу;
- б) компоненти залучені до роботи процесів;
- в) відповідальні за процеси.

Опишемо документи детальніше в таблиці 2.1.

Таблиця 2.1 – Реквізити документів вхідних даних

№	Найменування	Реквізити	Опис документу
1	Процеси для аналізу	назва; відповідальний за процес; дата і час початку; очікувана тривалість; дата і час завершення половини процесу; дата і час завершення 75% процесу; дата і час завершення процесу; компоненти, залучені до даного процесу	Містить список процесів і їх параметрів.
2	Компоненти залучені до роботи процесів	назва; складність компонента	Містить список компонент і їх параметрів.

## Продовження таблиці 2.1

№	Найменування	Реквізити	Опис документу
3	Відповідальні за процеси	ім'я; досвід роботи	Містить список користувачів і їх досвід.

**2.2 Вихідні дані**

Для розроблюваної системи вихідними даними є такі документи:

- прогнозовані значення часу завершення процесу і часу виконання частин процесу;
- список процесів у послідовному вигляді.

Опишемо документи детальніше в таблиці 2.2.

Таблиця 2.2 – Реквізити документів вхідних даних

№	Найменування	Реквізити	Опис документу
1	Прогнозовані значення часу завершення процесу і часу виконання частин процесу	назва процесу прогнозовані дата і час завершення половини процесу; прогнозовані дата і час завершення 75% процесу; прогнозовані дата і час завершення процесу	Результати прогнозування часу завершення частин процесу.
2	Список процесів у послідовному вигляді	дивитися реквізити процесів для аналізу (таблиця 2.1)	Табличне представлення послідовності процесів.

**2.3 Опис структури бази даних**

Наведемо перелік таблиць бази даних та їх полів у таблиці 2.3.

					ДП 6319.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

Таблиця 2.3 – Перелік таблиць та полів

№	Назва таблиці	Назва таблиці в базі даних	Поля
1	Процеси	proc	Id процесу; назва; дата і час початку процесу; очікувана тривалість; дата і час завершення половини процесу; дата і час завершення 75% процесу; дата і час завершення процесу; Id користувача; прогнозовані дата і час завершення половини процесу; прогнозовані дата і час завершення 75% процесу; прогнозовані дата і час завершення процесу
2	Користувачі	usr	Id користувача; ім`я; пароль; чи активний; досвід роботи

## Продовження таблиці 2.3

№	Назва таблиці	Назва таблиці в базі даних	Поля
3	Компоненти	component	Id компонента; назва; складність
4	Зв'язок між процесами	connection	Id зв'язку; Id процесу 1; Id процесу 2;
5	Процеси-компоненти	process_component	Id процесу (зовнішній ключ); Id компонента (зовнішній ключ)
6	Ролі користувачів	user_role	Id користувача; Роль

Схема бази даних наведена у графічному матеріалі. Розглянемо детальніше кожен з наведених таблиць.

Деталі таблиць та збереженої в них інформації наведено в таблицях 2.4 – 2.9.

Таблиця 2.4 – Опис полів таблиці «Процеси»

№	Назва поля	Тип поля	Семантика поля	Назва поля в БД
1	Id процесу	INT	Унікальний ідентифікатор процесу	id
2	Назва	VARCHAR(255)	Назва процесу	name
3	Дата і час початку процесу	DATETIME(6)	Дата і час початку процесу	start_date



Продовження таблиці 2.4

№	Назва поля	Тип поля	Семантика поля	Назва поля в БД
4	Очікувана тривалість	INT	Очікувана тривалість процесу (в годинах)	expected_duration
5	Дата і час завершення половини процесу	DATETIME(6)	Дата і час завершення половини процесу	half_over
6	Дата і час завершення 75% процесу	DATETIME(6)	Дата і час завершення 75% процесу	quater_left
7	Дата і час завершення процесу	DATETIME(6)	Дата і час завершення процесу	finish_date
8	Id користувача	BIGINT	Ідентифікатор відповідального за процес користувача	user_id
9	Прогнозовані дата і час завершення половини процесу	DATETIME(6)	Прогнозовані дата і час завершення половини процесу	predicted_half
10	Прогнозовані дата і час завершення 75% процесу	DATETIME(6)	Прогнозовані дата і час завершення 75% процесу	predicted_quater

Продовження таблиці 2.4

№	Назва поля	Тип поля	Семантика поля	Назва поля в БД
11	Прогнозовані дата і час завершення процесу	DATETIME(6)	Прогнозовані дата і час завершення процесу	predicted_finish

Таблиця 2.5 – Опис полів таблиці «Користувачі»

№	Назва поля	Тип поля	Семантика поля	Назва поля в БД
1	Id користувача	BIGINT	Унікальний ідентифікатор користувача (аналітика)	id
2	Ім'я	VARCHAR(255)	Ім'я користувача	username
3	Пароль	VARCHAR(255)	Пароль користувача для входу в систему	password
4	Чи активний	BIT(1)	Мітка активності користувача	active
5	Досвід роботи	INT	Досвід роботи користувача (роки)	workexperience

Таблиця 2.6 – Опис полів таблиці «Компоненти»

№	Назва поля	Тип поля	Опис	Назва поля в БД
1	Id компонента	Number (Число)	Унікальний ідентифікатор спринта в системі	id
2	Назва	VARCHAR(255)	Назва компонента	name

Продовження таблиці 2.6

№	Назва поля	Тип поля	Опис	Назва поля в БД
3	Складність	VARCHAR(255)	Складність компонента	complexity

Таблиця 2.7 – Опис полів таблиці «Зв'язки між процесами»

№	Назва поля	Тип поля	Опис	Назва поля в БД
1	Id зв'язку	INT	Унікальний ідентифікатор зв'язку між процесами	id
2	Id процесу 1	INT	Ідентифікатор процесу, із якого йде зв'язок в інший процес	process1_id
3	Id процесу 2	INT	Ідентифікатор процесу, в який іде зв'язок з іншого	process2_id

Таблиця 2.8 – Опис полів таблиці «Процеси-компоненти»

№	Назва поля	Тип поля	Опис	Назва поля в БД
1	Id процесу	Number (Число)	Ідентифікатор процесу	process_id
2	Id компонента	String (Рядковий тип даних)	Ідентифікатор компонента	component_id

Таблиця 2.9 – Опис полів таблиці «Ролі користувачів»

№	Назва поля	Тип поля	Опис	Назва поля в БД
---	------------	----------	------	-----------------

1	Id користувача	Number (Число)	Ідентифікатор користувача	user_id
2	Роль	Number (Число)	Роль користувача	roles

**Висновок до розділу**

У розділі «Інформаційне забезпечення» було описано загальну структуру вхідних даних, що потрібні для правильної роботи застосунку, а також вихідні дані, що отримуються в результаті прогнозування стану процесу та в результаті пошуку найдовшого ланцюга, що виходить з процесу.

Розроблено схему і модель бази даних з врахуванням усіх сутностей і зв'язків між ними, що подано у вигляді таблиць. Наведено детальний опис полів кожної таблиці із зазначенням типу поля та його семантичного значення.

### 3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

#### 3.1 Змістовна постановка задачі

##### 3.1.1 Змістовна постановка задачі 1

Процеси перебувають у зв'язках між собою, тобто деякі процеси повинні виконуватися раніше інших, що задано відношенням передування. Для певного процесу знайти найдовший ланцюг процесів, для якого він є початком.

##### 3.1.2 Змістовна постановка задачі 2

Існує набір процесів із параметрами, що їх визначають: очікувана тривалість, досвід роботи відповідального за процес, складність залучених компонент. У процесах визначено дату і час завершення 50% процесу, 75% процесу й усього процесу. На основі цих даних створити алгоритм, що буде прогнозувати час й дати завершення 50% процесу, 75% процесу й усього процесу для нових заданих процесів.

#### 3.2 Формалізована постановка задачі

##### 3.2.1 Формалізована постановка задачі 1

Нехай  $L = \{P, E\}$  – пара елементів, де  $P$  – множина процесів,  $E$  – множина зв'язків між процесами. Зв'язок задається парою процесів  $E = \{p_1, p_2\}$ , де  $p_1$  – процес, з якого починається зв'язок, а  $p_2$  – процес, на якому закінчується зв'язок. Для конкретного процесу  $p$  знайти послідовність процесів, що визначає найдовший ланцюг, у якому  $p$  – початок ланцюга.

##### 3.2.2 Формалізована постановка задачі 2

На основі  $X = \{d, e, c\}$  – трійки вхідних параметрів, де  $d$  – очікувана тривалість,  $e$  – досвід відповідального,  $c$  – залучені компоненти

					ДП 6319.00.000 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

спрогнозувати значення  $Y = \{t_1, t_2, t_3\}$  – трійки вихідних параметрів,  $t_1$  – прогнозована дата і час завершення 50% процесу,  $t_2$  – прогнозована дата і час завершення 75% процесу,  $t_3$  – прогнозовані дата і час завершення усього процесу.

### 3.3 Обґрунтування методів розв'язання

Для розв'язку задачі про знаходження найдовшого ланцюга процесів підходить алгоритм з використанням методу критичного шляху. Це обумовлено тим, що метод критичного шляху дозволяє знайти найдовший ланцюг в орієнтованому ациклічному графі, яким є граф процесів і зв'язків між ними [6].

Для прогнозування дати та часу завершення 50% процесу, 75% процесу й усього процесу для нових заданих процесів потрібен гнучкий метод. Часові ряди не підходять тому, що їх основна задача пошук тренду, що не відповідає потребі. Методи лінійного моделювання можуть не підійти, оскільки невідомий характер того, як потрібні дати залежать від вхідних параметрів. Нейронна мережа з методом зворотного поширення помилки при навчанні дозволяє отримати досить якісний результат.

### 3.4 Опис методів розв'язання

#### 3.4.1 Опис методу розв'язання задачі 1

Початок перегляду зв'язків між процесами починається зі зв'язків процесу, який буде початковим для шуканого ланцюга. Нехай черговий зв'язок, який можна переглянути, пов'язує два процеси. Якщо для першого процесу визначено можливий час його звершення і цей час плюс один більше можливого часу настання другого етапу, тоді для другого процесу встановлюється новий можливий час настання, що дорівнює очікуваному часу тривалості процесу плюс один. Розв'язок закінчується, коли черговий перегляд зв'язків не викликає жодного виправлення можливого значення

					ДП 6319.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

часу закінчення ланцюга. В результаті визначається процес з найбільш пізнім часом настання, що є останнім процесом ланцюга, і знайдений ланцюг першого процесу до його останнього буде найдовшим [7].

Головною функцією пошуку найдовшого ланцюга з процесу є функція з назвою «Пошук ланцюга».

Позначення:

- $s$  – процес;
- $first$  – початковий процес ланцюга;
- $last$  – умовний процес, до якого ідуть усі процеси, які не мають вихідних зв'язків;
- $v$  – зв'язок;
- $e(s)$  – час раннього початку процесу;
- $f(s)$  – час пізнього початку процесу;
- $v(from)$  – процес, у якому починається зв'язок;
- $v(to)$  – процес, у якому закінчується зв'язок;

### Пошук ланцюга

Крок 1. Встановлення раннього початку.

Крок 2. Встановлення пізнього початку.

Крок 3. Встановити ланцюг.

Крок 4.  $M = \{s: v(from) = first \wedge v(to) = s\}$  – множина етапів.

Крок 5. Поки не дійшли до етапу завершення процесу, то для всіх  $s \in M$  виконуються кроки 5.1-5.6.

Крок 5.1. Для всіх  $v \in V: v(to) = s$  виконуються кроки 5.2 – 5.6.

Крок 5.2.

Якщо  $e(v(to)) = f(v(to)) \wedge e(v(from)) = f(v(from)) \wedge f(s) = c(v) + f(v(from))$ , то крок 5.3, інакше – крок 5.1.

Крок 5.3. Додати  $v$  в ланцюг

Крок 5.4. Якщо  $s = last$ , то крок 6, інакше – крок 5.5.

Крок 5.5.  $M = \emptyset$

Крок 5.6.  $M = M \cup \{v(from) = s\}$

Крок 6. Кінець.

**Встановлення раннього початку:**

Крок 1.  $\forall s \in S: e(s) = -\infty; f(s) = +\infty$ .

Крок 2. Для початкового етапу:  $e(first) = 0$ .

Крок 3.  $M = \{s: v(from) = first \wedge v(to) = s\}$

Крок 4. Поки  $M \neq \emptyset$ , то для всіх  $s \in M$  виконувати кроки 4.1 – 4.5.

Крок 4.1. Для всіх  $v \in V: v(to) = s$  виконувати кроки 4.2 – 4.5.

Крок 4.2. Якщо  $e(v(from)) \neq -\infty$ , то крок 4.3, інакше – крок 4.

Крок 4.3.  $e(s) = \max\{e(s); e(v(from)) + 1\}$

Крок 4.4. Вилучити  $s$  з  $M$ ,  $M = M \setminus s$ .

Крок 4.5.  $M = M \cup \{v(from) = s\}$

Крок 5. Кінець.

**Встановлення пізнього початку:**

Крок 1. Для початкового етапу:  $f(first) = 0$ .

Крок 2. Для всіх  $v: v(to) = last, f(last) = \max\{e(v(from)) + 1; f(last)\}$

Крок 3.  $N = \{s: v(from) = s \wedge v(to) = last\}$

Крок 4. Поки  $N \neq \emptyset$ , то для всіх  $s \in N$  виконувати кроки 4.1 – 4.5.

Крок 4.1. Для всіх  $v \in V: v(from) = s$  виконувати кроки 4.2 – 4.5.

Крок 4.2. Якщо  $f(v(to)) \neq +\infty$ , то крок 4.3, інакше – крок 4.

Крок 4.3.  $f(s) = \min\{f(s); f(v(to)) - 1\}$

Крок 4.4. Вилучити  $s$  з  $N$ ,  $N = N \setminus s$ .

Крок 4.5.  $N = N \cup \{v(to) = s\}$

Крок 5. Кінець.

Для прикладу розглянемо зв'язки між процесами, що зображено на рисунку 3.1, знайдемо найдовший ланцюг, що починається з процесу 5. Таким ланцюгом є ланцюг 5-6-9-10-12-14-15-17-18-19.



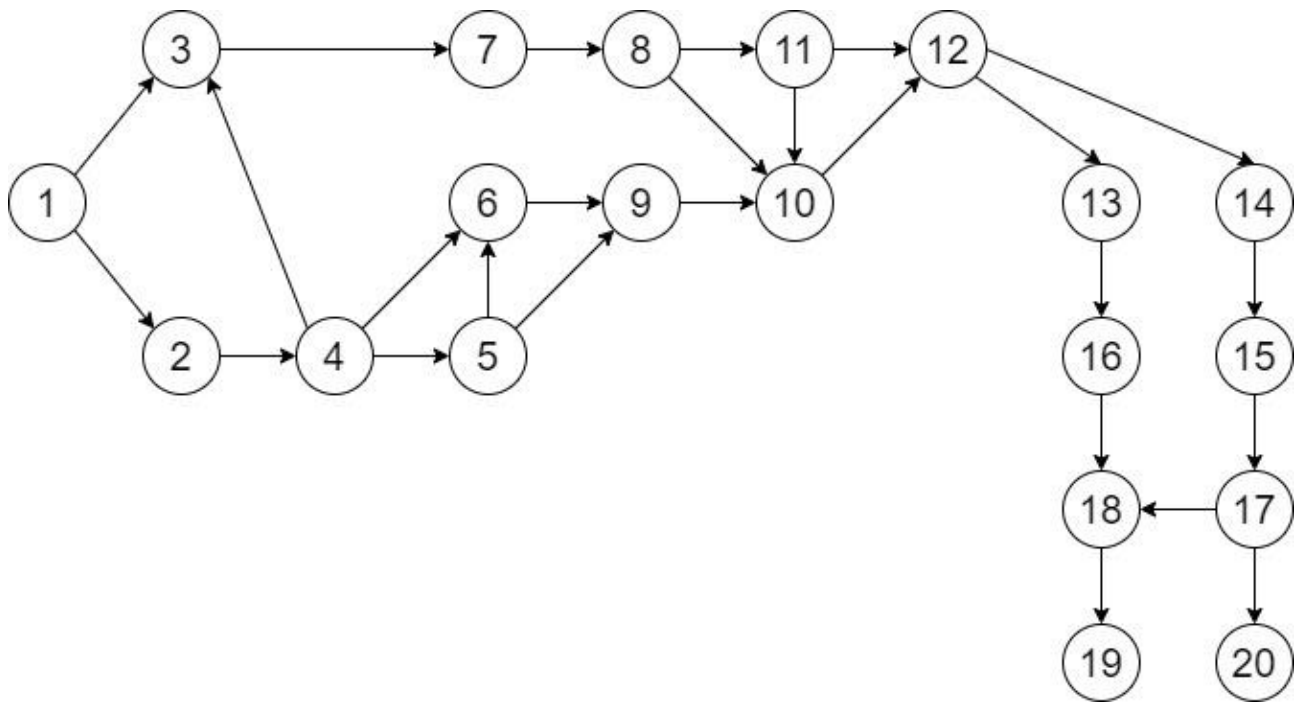


Рисунок 3.1 – Граф зв'язків між процесами

### 3.4.2 Опис методу розв'язання задачі 2

Для розв'язання задачі прогнозування дат завершення процесу пропонується використовувати нейронну мережу з використанням алгоритму навчання методом зворотного поширення помилки. Нейронна мережа – обчислювальна система, що складається з ряду простих, взаємопов'язаних елементів обробки, які обробляють інформацію шляхом їх динамічної реакції стану на зовнішні входи [8]. Подібність нейронної мережі до людського мозку полягає в таких аспектах:

Сигнали надходять в мережу із навколишнього середовища і використовуються в процесі навчання.

Для накопичення знань використовуються зв'язки між нейронами, що називаються синаптичними вагами [9].

Просту мережу можна описати так: у кожному нейроні вхідний сигнал обробляє активаційна функція. Вхідні параметри приходять в нейрони вхідного шару, оброблені сигнали передаються в усі нейрони прихованого шару, де їх зважена сума знову ж обробляється активаційною функцією [10].

Сигнал передається з кожного нейрона у вихідний шар, де відбуваються аналогічні операції. Проста нейромережа представлена на рисунку 3.2.

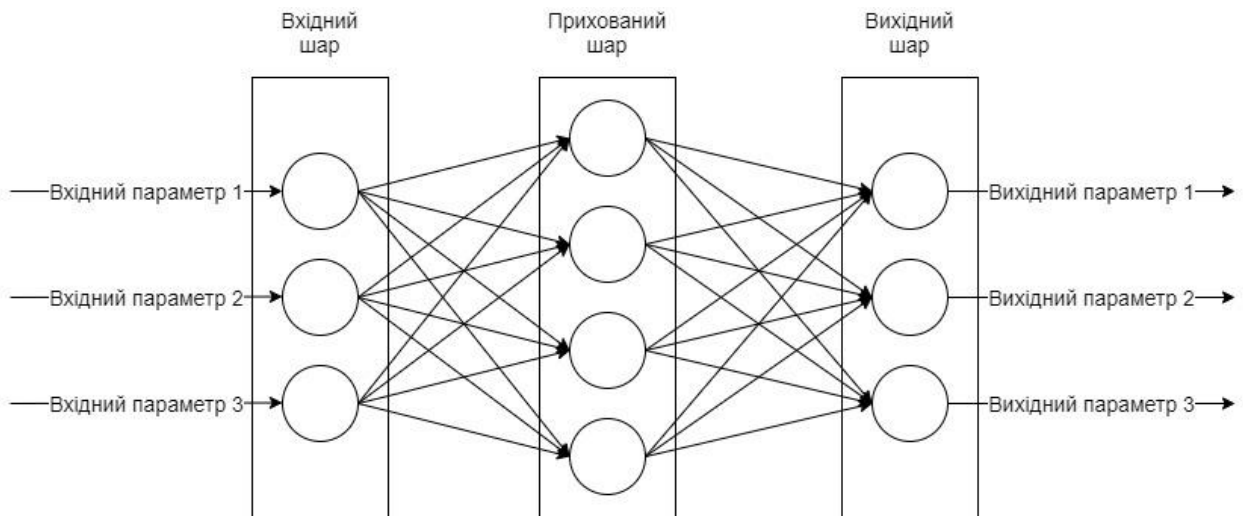


Рисунок 3.2 – Схематичне зображення нейронної мережі

Загалом робота нейронної мережі з використанням методу зворотного поширення помилки складається з таких етапів:

а) навчання

1) отримання вхідних даних з подальшим їх поширенням в напрямку виходів;

2) зворотне поширення відповідної помилки;

б) корекція ваг;

в) прогнозування – подача вхідних даних і поширенням в напрямку виходів.

**Алгоритм навчання методом зворотного поширення помилки:**

Крок 1. Випадкова початкова ініціалізація ваг  $v_{ij}$ ,  $w_{jk}$ , де  $i = 1 \dots n$ ,  $j = 1 \dots p$ ,  $k = 1 \dots t$ , причому  $n$  – кількість вхідних нейронів,  $p$  – кількість прихованих нейронів,  $t$  – кількість вихідних нейронів.

Крок 2. Поки не виконується умова припинення роботи алгоритму, то виконуємо кроки 3-10.

Крок 3. Для кожної пари вхідних і вихідних значень виконуються кроки 4-9.

Крок 4. Кожен вхідний нейрон  $X_i$  надсилає отриманий вхідний сигнал  $x_i$  усім нейронам у наступному прихованому шарі.

Крок 5. Кожен прихований нейрон  $Z_j$  сумує зважені вхідні сигнали:  $z_j^{(in)} = v_{0j} + \sum_i x_i * v_{ij}$ . Застосовується активаційна функція:  $z_j = f(z_j^{(in)})$ . Нейрон надсилає значення  $z_j$  на наступний шар.

Крок 6. Кожен вихідний нейрон  $Y_k$  сумує зважені вхідні сигнали:  $y_k^{(in)} = w_{0k} + \sum_j z_j * v_{jk}$ . Застосовується активаційна функція:  $y_k = f(y_k^{(in)})$ .

Крок 7. Кожен вихідний нейрон  $Y_k$  отримує цільове значення  $t_k$  – вихідне значення для вхідного сигналу, що надійшов. Тоді нейрон обраховує помилку  $\sigma_k = \frac{t_k - y_k}{f'(y_k^{(in)})}$ . Ще знаходиться величина, на яку зміниться вага зв'язку  $w_{jk}$ ,  $\Delta w_{jk} = a * \sigma_k * z_j$ . Знаходиться величина корекції зміщення:  $\Delta w_{0k} = a * \sigma_k$ . Нейрон надсилає  $\sigma_k$  нейронам попереднього шару ( $a$  – швидкість навчання).

Крок 8. Кожен прихований нейрон  $Z_j$  знаходить суму вхідних помилок, що отримані з нейронів наступного шару  $\sigma_j^{(in)} = \sum_k \sigma_k * w_{jk}$  і знаходить величину помилки  $\sigma_j = \sigma_j^{(in)} * f'(z_j^{(in)})$ . Також знаходиться величина на яку зміняться ваги  $v_{ij}$ ,  $\Delta v_{ij} = a * \sigma_j * x_i$ . Знаходиться величина корекції зміщення:  $\Delta v_{0j} = a * \sigma_j$ .

Крок 9. Кожен вхідний нейрон  $X_i$  змінює ваги своїх зв'язків з елементом зміщення і прихованими нейронами  $w_{jk}^{(new)} = w_{jk}^{(old)} + \Delta w_{jk}$ .

Кожен прихований нейрон  $Z_j$  змінює ваги своїх зв'язків з елементом зміщення і вихідними нейронами  $v_{ij}^{(new)} = v_{ij}^{(old)} + \Delta v_{ij}$ .

Крок 10. Перевірка умов припинення роботи алгоритму [11].

### Висновок до розділу

У розділі «Математичне забезпечення» були наведені змістовні та формалізовані постановки задач для пошуку найдовшого ланцюга процесів та для прогнозування часу і дат завершення процесу.

У підрозділі змістовної постановки вказано особливості графа для пошуку найдовшого ланцюга процесів та вказано вхідні та вихідні параметри для задачі прогнозування плин timer процесів. У підрозділі формалізованої постановки надано постановку задачі з використанням математичних позначень.

Вказано переваги обраних методів розв'язку задач над іншим, зокрема чому для прогнозування плин timer процесів не підходять методи часових рядів та лінійне моделювання.

У підрозділах опису методів розв'язання наведені алгоритми пошуку найдовшого ланцюга процесів, що починається з вказаного процесу, і алгоритм навчання нейронної мережі методом зворотного поширення помилки.

## 4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

### 4.1 Засоби розробки

Щоб розробити програмний застосунок обрано мову програмування Java. Java має власну структуру, правила синтаксису та використовувану парадигму програмування. Java базується на концепції об'єктно-орієнтованого програмування, яку підтримують усі функції мови. Структурно мова Java починається з пакетів, що є механізмом простору імен мови. У середині пакетів розміщені класи, а в середині класів використовуються методи, змінні, константи та інші структурні одиниці програмного коду.

Переглядаючи переваги та недоліки мови Java, можна із впевненістю сказати, що кількість перших значно вища. До переваг мови Java посеред іншого належать такі:

- Java пропонує широку функціональність та мобільність, бо застосунки написані на одні платформі, можуть бути поширені на настільних комп'ютерах, мобільних пристроях, вбудованих системах;
- Java є безкоштовною, простою у використанні, об'єктно-орієнтованою, розподіленою мовою, що підтримує багатопоточність та надає широкі можливості використання мультимедіа та мережі;
- Java – мова з уже серйозним історичним підґрунтям, тому більш стабільна та передбачувана в порівнянні з новішими мовами;
- бібліотека класів Java забезпечує можливість працювати на багатьох програмно-апаратних платформах;
- Java має чи не найбільшу активну спільноту користувачів і доступну підтримку, що дозволяє знайти в мережі відповіді на усі запитання, що виникають в процесі розробки;
- на відміну від C і C++, застосунки на Java компілюються незалежно від платформи мовою байт-коду, тому одну і ту ж програму можна

					ДП 6319.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

запускати на усіх комп'ютерах, де встановлена JVM (Java Virtual Machine);

- відносно проста пряма сумісність від однієї версії до іншої;
- Наявні потужні інструменти розробки, такі як: IntelliJ IDEA, Eclipse SDK і Netbeans, у яких можна легко виконати налагодження застосунку [12].

Для розробки застосунку використовувалось єдине середовище розробки – IntelliJ IDEA. Ключовими функціями цього середовища є допомога в кодуванні, інтелектуальна навігація та пошук, різноманітні рефакторинги, аналіз коду. Величезним плюсом є підтримка веброзробки, адже систему розроблено у вигляді вебзастосунку. Інтелектуальне завершення та завершення ланцюга видає перелік допустимих символів в конкретному контексті, що теж спрощує роботу. Вбудований у середовище розробки налагоджувач показує значення змінних просто поруч їх використання у вихідному коді. Значною перевагою є те, що IntelliJ IDEA підтримує інструменти зборки, що легко інтегруються та дозволяють автоматизувати компіляцію, упаковку та розгортання застосунку. Також IntelliJ IDEA підтримує основні сервери застосувань, такі як Tomcat, JBoss, WebLogic та інші. Можна розгорнути артефакти на серверах застосунків і налагодити просто з середовища IDE.

Для розробки застосунку використано фреймворк Spring Boot. Spring Boot – фреймворк із відкритим вихідним кодом. Він надає розробникам Java платформу для початку роботи з застосунком Spring, що конфігурується автоматично. Spring Boot побудований поверх середовища Spring і надається з множиною залежностей, що можна під'єднати до Spring застосунку.

Прикладами Spring є Spring Kafka, Spring Web Services і Spring Web Security. Проте розробник має самостійно налаштовувати кожен елемент, користуючись багатьма файлами конфігурацій XML чи анотацій. Spring фреймворк орієнтований на забезпечення гнучкості завдяки функції введення

залежностей. Це допомагає швидко додавати потрібну залежність в слабо зв'язній формі (якнайменше зв'язків між компонентами застосунку). До інших переваг Spring належать:

- легкий каркас;
- модульність архітектури дозволяє обирати зайві деталі та вилучати їх;
- має підтримку конфігурації XML та анотацій;
- надає абстракцію програмного забезпечення об'єктно-реляційного маппера (ORM) для розробки логіки персистентності ORM;
- сумісний з багатьма сервісами проміжного програмного забезпечення;
- підтримує інфраструктуру JDBC, що підвищує продуктивність та зменшує кількість помилок [13].

У свою ж чергу Spring Boot орієнтований на скорочення коду і надання простого способу запустити Spring застосунок. До важливих особливостей Spring Boot належать:

- а) автоконфігурація – розробники можуть автоматично налаштовувати Spring застосунок. Проте Spring Boot може також змінювати конфігурацію в залежності від вказаних залежностей. Якщо вказується MySQL як залежність застосунку, то в конфігурацію буде додано MySQL з'єднувач. Для задання користувацької конфігурації потрібно перевизначити конфігурацію за замовчуванням;
- б) автономність – немає потреби розгортати застосунок на вебсервері. Просто вводиться команда запуску, що стартувати застосунок;
- в) огляд – Spring Boot самостійно вирішує, які параметри використовувати для конфігурації за замовчуванням. Наприклад, якщо у стартову версію pom файлу Spring Boot додати залежність JPA, то в пам'яті автоматично буде сконфігуровано базу даних, менеджер об'єктів гібернації та просте джерело даних. Налаштування Spring Boot дозволяє розробникам швидко починати роботу над проєктами;

Наведені вище особливості можна вважати перевагами Spring Boot, до яких також можна додати такі:

- наявність великої кількості доступних навчальних матеріалів та курсів;
- скорочення часу розробки та підвищення загальної продуктивності команди;
- спрощення створення та тестування застосунків на основі Java, шляхом налаштувань по замовчуванню для модульних та інтеграційних тестів;
- дозволяє уникнути великої кількості стандартного коду, анотацій та конфігурацій XML;
- надається з вбудованими HTTP-серверами, такими як Tomcat чи Jetty.
- додає набір плагінів, що розробники можуть використовувати для роботи з вбудованими чи серверними базами даних. Spring Boot дозволяє легко підключатися до баз даних і служб черг, як от: Oracle, PostgreSQL, MySQL, MongoDB, Solr, Rabbit MQ та інших [14].

Для шаблонізації вебсторінок використано FreeMarker, що є механізмом шаблонів з відкритим кодом для генерації текстового виводу. Ці шаблони написані на власній мові FreeMarker, що є простою спеціалізованою мовою. FreeMarker показує підготовані мовою програмування дані з використанням шаблону. В межах шаблону потрібно зосереджуватися на тому, як представляти дані, а не які дані представляти. Особливістю FreeMarker є те, що він володіє потужною мовою шаблонів: умовні блоки, ітерації, призначення, рядкові й арифметичні операції та форматування, макроси та функції – усе це підтримується FreeMarker. Шаблонізатор є водночас і багатоцільовим і легким – нульові залежності, будь-який формати виводу, велика кількість параметрів конфігурації. FreeMarker чутливий до



локальних представлень і форматування дати і часу. Модель даних для FreeMarker – універсальна, об'єкти Java надаються шаблону у вигляді дерева змінних через адаптери підключення.

Для налаштування і покращення зовнішнього вигляду вебсторінок використано фреймворк Bootstrap. Bootstrap – фреймворк орієнтований на інтерфейс, безкоштовний для використання у розробці вебсайтів та застосунків. Вебсайти, розроблені з використанням Bootstrap сумісні й з Windows, і з Android, і з IOS. Bootstrap містить велику кількість готових шаблонів зовнішнього вигляду сторінок. Bootstrap має ряд якісних особливостей, а саме:

- легкий для розуміння;
- просте налаштування, Bootstrap спроектований в адаптивній сітці, макетах і компонентах із 12 стовпцями, його легко налаштувати, можна зробити сітку фіксованою чи виконати зміщення і вкладання стовпців;
- службові класи дозволяють налаштувати зображення чи приховання певного фрагмента контенту на пристроях, в залежності від розміру використовуваного екрана;
- попередньо вбудовані компоненти, як от: випадне меню, кнопки, навігація, сповіщення, індикатори [15].

Для зберігання даних, що використовуються та продукуються системою використано базу даних MySQL. MySQL – реляційна система управління базами даних з відкритим вихідним кодом. У її основі – SQL, що використовується для додання, видалення та зміни інформації в базі даних [16]. MySQL – стабільна, надійна та потужна система управління базами даних, що володіє такими особливостями:

- а) безпека даних. MySQL – відома як найбільш безпечна та надійна СУБД, що використовується в таких веб застосунках, як Facebook, Twitter та інших;

- б) масштабованість за потребою. MySQL пропонує велику масштабованість, щоб спростити керування вбудованими застосунками, шляхом використання малого обсягу навіть у великих сховищах на терабайти даних;
- в) висока продуктивність. MySQL має таку структуру механізму збереження, що допомагає системним адміністраторам налаштовувати сервер бази даних до якнайкращої продуктивності;
- г) цілодобова доступність. MySQL пропонує широкий спектр рішень високої доступності, як от спеціалізовані кластерні сервери та конфігурації реплікації master/slave;
- д) комплексна підтримка транзакцій забезпечується шляхом повної атомарної, погодженої, ізольованої, довготермінової підтримки транзакцій, підтримки декількох версій транзакцій. Ще доступне необмежене блокування на рівні рядків, що є ідеальним рішенням для повної цілісності даних;
- е) контроль робочого процесу – середній час завантаження й установа бази даних не залежить від використовуваної платформи;
- ж) гнучкість відкритого вихідного коду підтверджується постійною безпечною обробкою та цілодобовою підтримкою MySQL. Це допомагає у швидкому та якісному технічному обслуговуванні [17].

Для успішного зіставлення сутностей бази даних з об'єктами програми використовувався фреймворк Hibernate. Hibernate – фреймворк з відкритим вихідним кодом, що забезпечує зіставлення об'єктно-орієнтованих моделей предметної області із реляційними базами даних для веб застосунків. Зіставлення об'єктів засновано на контейнеризації об'єктів та абстракції, що забезпечує цю місткість. Абстракція дозволяє звертатись до об'єктів і при цьому не задумуватися як вони зв'язані з джерелом даних. Інфраструктура Hibernate ORM забезпечує зіставлення класів Java з таблицями бази даних і забезпечує запити й пошук. Усі виконані зміни інкапсуються в самому

джерелі даних, тому при зміні цих чи їх API застосункам, що використовують ORM, не потрібно цього [18]. Аналогічно і розробники можуть мати погоджене уявлення про об'єкти зі зміною часу, хоча джерела, що їх доставляють, приймачі, що отримують і застосунки, що отримують доступ, можуть змінитися.

## 4.2 Вимоги до технічного забезпечення

Для правильної роботи розробленого застосунку до складу технічних засобів мають входити:

а) комп'ютер із такою конфігурацією:

- 1) процесор з тактовою частотою не нижче 1.6 ГГц;
- 2) достатній об'єм оперативної пам'яті (не менше 1 ГБ);
- 3) інші складові можуть мати будь-які параметри, оскільки вони впливають на роботу програми неістотно;

б) також має бути встановлене таке програмне забезпечення:

- 1) операційна система Windows 7 або вище;
- 2) СУБД MySQL 8.0 або вище;
- 3) .NET Framework 4.4 або вище;
- 4) JDK 8 або вище;

в) комп'ютерна периферія, а саме:

- 1) монітор;
- 2) маніпулятор-миша;
- 3) клавіатура.

## 4.3 Архітектура програмного забезпечення

Розроблений вебзастосунок виконано із використанням архітектурного шаблону MVC (Модель-вид-контролер). Ключовий принцип MVC – у кожного фрагменту коду є свої цілі, і, як правило, вони відрізняються. Фрагменти коду контролюють дані застосунку, інші ж – його зовнішній

вигляд, ще частина коду – його функціональність. MVC дозволяє грамотно розмістити блоки коду, їх структурувати та зробити зручним повторне використання коду. Ще MVC допомагає в полегшенні використання різними елементами один одного. Частина MVC мають таке призначення:

- модель – зберігає необроблені дані чи визначає основні елементи застосування;
- вид – напряду взаємодіє користувач застосунку;
- контролер – зв'язка між моделлю та видом, контролер отримує введення від користувача і вирішує, як саме його опрацювати.

MVC надає стартову точку для втілення ідеї в код і, водночас, полегшує повернення від абстракції на рівень коду, бо можна визначити, який код що виконує. Окрім того, MVC – стандартизує код, і його легше зрозуміти іншим розробникам. MVC дозволяє вносити зміни в компоненти за бажанням, при цьому інші компоненти не потребуватимуть змін. Ще даний архітектурний шаблон дозволяє паралельно працювати над частинами застосунку з мінімальною залежністю між розробниками, що відповідають за різні частини [19].

Розроблений в рамках даної роботи застосунок відповідає архітектурі MVC, він складається з пакетів, що його формують. У таблиці 4.1 наведено всі пакети застосунку з поясненням їх призначення.

Таблиця 4.1 – Пакети розробленого застосунку

Назва пакету	Призначення пакету
config	Збереження конфігурації для доступу до застосунку, в тому числі авторизації.
controller	Збереження контролерів, що виконують функцію посередника між видом та моделлю.
CP	Збереження класів для пошуку найдовшого ланцюга.

Продовження таблиці 4.1

domain	Зберігає об'єкти, що відповідають сутностям бази даних.
NN	Збереження класів для роботи з нейронною мережею (прогнозування процесів).
repos	Збереження репозиторіїв для доступу до сутностей бази даних.
service	Збереження сервісу, що реалізує метод пошуку користувача за його іменем.

Отже, можна сказати, що за елемент MVC Модель відповідає пакет domain, за елемент Controller – пакет controller, за view відповідають ресурси застосунку, представлені .ftl файлами, що забезпечують генерацію файлів текстового виводу HTML.

#### 4.3.1 Діаграма класів

Загалом розроблений застосунок складається з багатьох класів та інтерфейсів. Фрагмент діаграми класів, що зображає класи пакетів controller та repos подано у графічному матеріалі.

#### 4.3.2 Діаграма послідовності

Щоб якнайкраще зрозуміти внутрішні процеси обміну даними між класами розглянемо детально як відбувається пошук найдовшого ланцюга, що починається в певному процесі. Розглянемо що відбувається при створенні нового процесу.

У частині графічного матеріалу надано схему послідовності взаємодії частин застосунку при пошуку найдовшого ланцюга, що починається в обраному процесі. Також надано схему послідовності для додання нового процесу.

### 4.3.3 Діаграма компонентів

Діаграму компонентів подано у графічному матеріалі. З неї видно, що застосунок складається з таких компонентів:

- вебсторінки – з ними працює користувач застосунку;
- база даних – зберігає інформацію, що потрібна для роботи застосунку;
- застосунок Spring Boot – відповідальний за всю роботу програми, отримує команди з вебсторінок та звертається до бази даних, щоб отримати інформацію.

### 4.3.4 Специфікація функцій

Усього кількість методів, що використовуються в розробленому застосунку, перевищує 160. Їх усіх описувати не є доцільно. Тому пропонується розглянути специфікацію функцій, що пов'язані з безпосередньою роботою користувача застосунку, тобто функціями з пакета controller. Специфікацію функцій наведено в таблиці 4.2.

Таблиця 4.2 – Специфікація функцій пакету controller

Клас	Прототип функції	Семантика функції	Параметри функції	Семантика параметрів
MainController	@GetMapping("/") public String greeting()	Повертає головну сторінку застосунку	-	-

Продовження таблиці 4.2

Клас	Прототип функції	Семантика функції	Параметри функції	Семантика параметрів
MainController	<pre>@GetMapping("main") public String main (@RequestParam (required = false, defaultValue = "")) String filter, Model model)</pre>	Повертає сторінку застосунку «Процеси»	<pre>(@RequestParam (required = false, defaultValue = "")) String filter: Model model</pre>	Запитуваний параметр для фільтрації процесів за назвою; держатель для атрибутів моделі
MainController	<pre>@GetMapping ("main/{proc}") public String findPath (@PathVariable Proc proc, Model model)</pre>	Повертає сторінку із знайденим найдовшим ланцюгом, що починається в обраному процесі	<pre>@PathVariable Proc proc; Model model</pre>	Процес для пошуку; держатель для атрибутів моделі

Продовження таблиці 4.2

Клас	Прототип функції	Семантика функції	Параметри функції	Семантика параметрів
MainController	<pre>@PostMapping("main") public String makePrediction (@RequestParam (required = false, defaultValue = "") String filter, @RequestParam (required = true, defaultValue = "10") Integer trainSize, Model model)</pre>	Повертає сторінку «Процеси» з виконаним прогнозуванням дати і часу завершення процесів	<pre>@RequestParam (required = false, defaultValue = "") String filter; @RequestParam (required = true, defaultValue = "10") Integer trainSize; Model model</pre>	Запитуваний параметр для фільтрації процесів за назвою; розмір навчальної вибірки; держатель для атрибутів моделі
ComponentController	<pre>@GetMapping("/components") public String allComponents (Model model)</pre>	Повертає сторінку «Компоненти»	Model model	Держатель для атрибутів моделі



Продовження таблиці 4.2

Клас	Прототип функції	Семантика функції	Параметри функції	Семантика параметрів
ComponentController	@PostMapping("/components") public String addComponent (@RequestParam String name, @RequestParam String complexity)	Повертає сторінку «Компоненти» з доданим новим компонентом	@RequestParam String name; @RequestParam String complexity	Назва компонента; Складність компонента
ProcessController	@GetMapping("/addProcess") public String addForm (Model model)	Повертає сторінку «Додання процесу»	Model model	Держатель для атрибутів моделі

Продовження таблиці 4.2

Клас	Прототип функції	Семантика функції	Параметри функції	Семантика параметрів
Process Controller	@PostMapping("/addProcess") public String addProcess( @AuthenticationPrincipal User user, @RequestParam String name, @RequestParam("startDate")@DateTimeFormat(iso = DateTimeFormat.ISO.DATE) Date startDate, @RequestParam Integer expectedDuration, @RequestParam Long[] compids)	Повертає сторінку «Процеси» з доданим новим процесом	@AuthenticationPrincipal User user; @RequestParam String name; @RequestParam("startDate")@DateTimeFormat(iso = DateTimeFormat.ISO.DATE) Date startDate; @RequestParam Integer expectedDuration; @RequestParam Long[] compids	Користувач, що відповідає за процес; назва процесу; дата початку; очікувана тривалість процесу; масив ідентифікаторів, пов'язаних компонентів
UserController	@GetMapping public String userList (Model model)	Повертає сторінку «Список користувачів»	Model model	Держатель для атрибутів моделі

Продовження таблиці 4.2

Клас	Прототип функції	Семантика функції	Параметри функції	Семантика параметрів
UserController	@GetMapping("/{user}")  public String userEditForm (@PathVariable User user, Model model)	Повертає сторінку «Редагування користувача»	@PathVariable User user;  Model model	Користувач; держатель для атрибутів моделі
UserController	@PostMapping  public String userSave (@RequestParam String username, @RequestParam Integer workexperience, @RequestParam Map<String, String> form, @RequestParam ("userId") User user)	Повертає сторінку «Список користувачів» з відредагованим користувачем	@RequestParam String username;  @RequestParam Integer workexperience;  @RequestParam Map<String, String> form;  @RequestParam ("userId") User user	Нове ім'я користувача; досвід роботи користувача; форма, що зберігає ролі користувача; користувач
RegistrationController	@GetMapping("/login")  public String login()	Повертає сторінку «Авторизація»	-	-

Продовження таблиці 4.2

Клас	Прототип функції	Семантика функції	Параметри функції	Семантика параметрів
RegistrationController	@GetMapping("/registration") public String registration()	Повертає сторінку «Реєстрація»	-	-
RegistrationController	@PostMapping("/registration") public String addUser(User user, Map<String, Object> model)	Повертає результат після намагання додати нового користувача	User user; Map<String, Object> model	Користувач; держатель для атрибутів моделі, що може повернути інформацію, що користувач вже наявний

**Висновок до розділу**

У розділі «Програмне та технічне забезпечення» розглянуто засоби розробки та технології, що використовувались при створенні програмного застосунку, а саме: мова програмування Java, інтегроване середовище розробки IntelliJ IDEA, фреймворк Spring Boot, шаблонізатор FreeMarker, фреймворк Bootstrap, реляційна СУБД MySQL, фреймворк Hibernate. Наведено обґрунтування такого вибору, зокрема вказані переваги даних засобів.

У підрозділі «Архітектура програмного забезпечення» розглянуто архітектуру застосунку, зокрема як для його побудови використано шаблон MVC, наведено таблицю пакетів, з яких складається застосунок.

У підрозділі «Специфікація функцій» детально описано функції пакету «Контролер», з яким безпосередньо взаємодіє користувач застосунку, в тому числі наведено семантику функцій та їх параметрів.

					ДП 6319.00.000 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

## 5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

### 5.1 Керівництво користувача

Щоб розпочати роботу з розробленим застосунком користувач повинен відкрити браузер і перейти за адресою <http://localhost:8080/>. Розглянемо детально, що побачить користувач після завантаження сторінки. Домашню сторінку зображено на рисунку 5.1.



Привіт, користувачу!

Це система аналізу та прогнозування інформаційно-комунікаційних процесів

Рисунок 5.1 – Домашня сторінка застосунку

Зокрема неавторизований користувач системи може бачити такі елементи:

- кнопка 1, «Перехід на домашню сторінку» – на ній показується привітання для користувача системи, саме на ній зараз знаходиться неавторизований користувач.
- кнопка 2, «Увійти в систему» – для авторизації у системі.

Натиснемо кнопку 2 і перейдемо на сторінку авторизації. Детально сторінку авторизації представлено на рисунку 5.2.

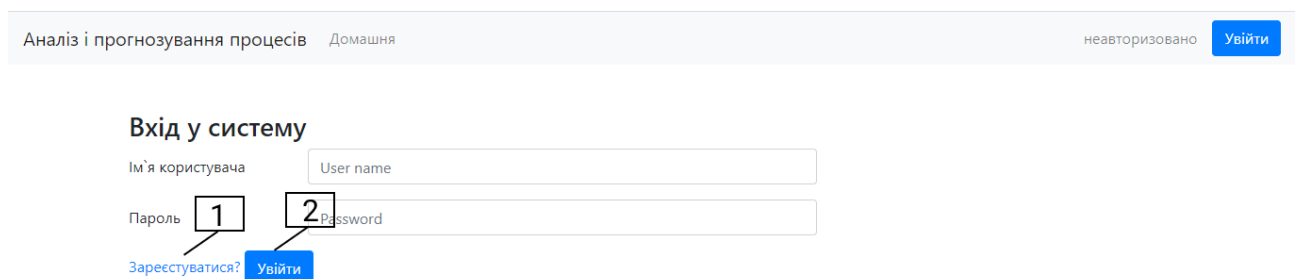


Рисунок 5.2 – Сторінка авторизації

Із рисунку видно, що на сторінці авторизації користувач може виконати такі дії:

					ДП 6319.00.000 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

- кнопка 1, «Зареєструватися», у випадку якщо він ще не зареєстрований;
- кнопка 2, «Увійти», попередньо ввівши своє ім'я та пароль.

Перейдемо на сторінку реєстрації, яку можна побачити на рисунку 5.3.

Рисунок 5.3 – Сторінка реєстрації

На сторінці реєстрації представлена одна ключова дія – зареєструватися, що є доступною після вводу імені та паролю за натиском на кнопку 1. Виконавши ці дії повертаємося на сторінку авторизації, на якій вводимо дані щойно зареєстрованого користувача. Натиснувши кнопку «Увійти» показано сторінку 1 для авторизованого користувача. Її вигляд представлено на рисунку 5.4.

**Привіт, користувачу!**

Це система аналізу та прогнозування інформаційно-комунікаційних процесів

Рисунок 5.4 – Домашня сторінка авторизованого користувача

На домашній сторінці авторизованого користувача на навігаційній панелі є доступними такі нові кнопки:

- кнопка 1, «Процеси» – перехід на сторінку для перегляду наявних процесів та виконання операцій з ними.
- кнопка 2, «Вийти» – кнопка для виходу із системи зареєстрованого користувача.

Виконаємо перехід на сторінку «Процеси» за кнопкою 1. Детально сторінку процеси зображено на рисунку 5.5.

					ДП 6319.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

Аналіз і прогнозування процесів Домашня Процеси user7 Вийти

## Процеси

Пошук процесу за назвою **1** Шукати **2** Додати новий процес

Розмір навчальної вибірки 10 **3** Зробити прогноз

Назва процесу	Відповідальний	Час початку	Очікувана тривалість (годин)	Завершено 50%	Прогноз 50%	Завершено 75%	Прогноз 75%	Дата завершення	Прогноз 100%	Залучені компоненти	<b>4</b> Знайти ланцюг
Process 1	user5	20 жовт. 2019 р., 16:25:00	15	21 жовт. 2019 р., 00:28:52	-	21 жовт. 2019 р., 02:32:02	-	21 жовт. 2019 р., 08:32:43	-	Component 4	Знайти ланцюг
Process 2	user2	20 жовт. 2019 р., 17:00:00	6	20 жовт. 2019 р., 21:19:20	-	20 жовт. 2019 р., 22:26:34	-	21 жовт. 2019 р., 01:19:27	-	Component 4	Знайти ланцюг
Process 3	user1	21 жовт. 2019 р., 08:40:00	10	21 жовт. 2019 р., 13:19:22	-	21 жовт. 2019 р., 17:30:47	-	21 жовт. 2019 р., 20:46:20	-	Component 1, Component 5	Знайти ланцюг
Process	user3	21 жовт.	8	21 жовт.	-	21 жовт.	-	21 жовт.	-	Component	Знайти

Рисунок 5.5 – Сторінка «Процеси»

Серед нових можливостей, що доступні на даній сторінці, є дії за такими кнопками:

- кнопка 1, «Шукати» – для пошуку процесу у списку за його назвою;
- кнопка 2, «Додати новий процес» – для додання нового процесу у список;
- кнопка 3, «Зробити прогноз» – викликає дію прогнозування часу і дати завершення процесу;
- кнопка 4, «Знайти ланцюг» – розміщена біля кожного процесу і викликає дію знаходження найдовшого ланцюга, що починається саме в обраному процесі.

Введемо назву процесу у поле для пошуку і натиснемо кнопку 1. На рисунку 5.6 можна пересвідчитися, що знайдено процес із заданою назвою.



## Процеси

Process 21  [Додати новий процес](#)

Розмір навчальної вибірки

Назва процесу	Відповідальний	Час початку	Очікувана тривалість (годин)	Завершено 50%	Прогноз 50%	Завершено 75%	Прогноз 75%	Дата завершення	Прогноз 100%	Залучені компоненти	
Process 21	Yurii	12 трав. 2020 р., 03:00:00	7	-	-	-	-	-	-		<a href="#">Знайти ланцюг</a>
Назва процесу	Відповідальний	Час початку	Очікувана тривалість (годин)	Завершено 50%	Прогноз 50%	Завершено 75%	Прогноз 75%	Дата завершення	Прогноз 100%	Залучені компоненти	

Рисунок 5.6 – Сторінка «Процеси» з процесами відфільтрованими за назвою

Продовжимо роботу на сторінці «Процеси», задавши розмір навчальної вибірки та натиснувши кнопку 3 «Зробити прогноз». Як це виглядає можна побачити на рисунку 5.7.

## Процеси

Пошук процесу за назвою  [Додати новий процес](#)

Розмір навчальної вибірки

Назва процесу	Відповідальний	Час початку	Очікувана тривалість (годин)	Завершено 50%	Прогноз 50%	Завершено 75%	Прогноз 75%	Дата завершення	Прогноз 100%	Залучені компоненти	
Process 1	user5	20 жовт. 2019 р., 16:25:00	15	21 жовт. 2019 р., 00:28:52	-	21 жовт. 2019 р., 02:32:02	-	21 жовт. 2019 р., 08:32:43	-	Component 4	<a href="#">Знайти ланцюг</a>
Process 2	user2	20 жовт. 2019 р., 17:00:00	6	20 жовт. 2019 р., 21:19:20	-	20 жовт. 2019 р., 22:26:34	-	21 жовт. 2019 р., 01:19:27	-	Component 4	<a href="#">Знайти ланцюг</a>
Process 3	user1	21 жовт. 2019 р., 08:40:00	10	21 жовт. 2019 р., 13:19:22	-	21 жовт. 2019 р., 17:30:47	-	21 жовт. 2019 р., 20:46:20	-	Component 1, Component 5	<a href="#">Знайти ланцюг</a>
localhost...				21 жовт.	-	21 жовт.	-	21 жовт.	-	Component	<a href="#">Знайти</a>

Рисунок 5.7 – Сторінка «Процеси» перед виконанням прогнозування

Результати прогнозування можна побачити на рисунку 5.8.

Process 15	user2	24 жовт. 2019 р., 19:40:00	5	24 жовт. 2019 р., 23:04:06	24 жовт. 2019 р., 22:38:29	25 жовт. 2019 р., 00:52:09	25 жовт. 2019 р., 00:38:48	25 жовт. 2019 р., 01:56:11	25 жовт. 2019 р., 02:03:02	Component 4	<a href="#">Знайти ланцюг</a>
Process 16	user4	25 жовт. 2019 р., 09:00:00	8	25 жовт. 2019 р., 12:44:38	25 жовт. 2019 р., 12:06:42	25 жовт. 2019 р., 14:15:22	25 жовт. 2019 р., 14:10:15	25 жовт. 2019 р., 16:46:34	25 жовт. 2019 р., 15:58:10	Component 1	<a href="#">Знайти ланцюг</a>
Process 17	user3	26 жовт. 2019 р., 09:00:00	4	26 жовт. 2019 р., 11:25:44	26 жовт. 2019 р., 11:02:45	26 жовт. 2019 р., 13:02:53	26 жовт. 2019 р., 12:24:39	26 жовт. 2019 р., 13:57:32	26 жовт. 2019 р., 13:41:45	Component 5	<a href="#">Знайти ланцюг</a>
Process 18	user2	26 жовт. 2019 р., 10:50:00	5	26 жовт. 2019 р., 13:13:14	26 жовт. 2019 р., 13:16:04	26 жовт. 2019 р., 14:31:41	26 жовт. 2019 р., 15:03:56	26 жовт. 2019 р., 16:13:59	26 жовт. 2019 р., 16:01:05	Component 2	<a href="#">Знайти ланцюг</a>
Process 19	user2	26 жовт. 2019 р., 12:40:00	10	26 жовт. 2019 р., 16:50:34	26 жовт. 2019 р., 17:38:03	26 жовт. 2019 р., 19:39:41	26 жовт. 2019 р., 20:59:26	27 жовт. 2019 р., 00:02:47	26 жовт. 2019 р., 23:01:51	Component 1	<a href="#">Знайти ланцюг</a>
Process 20	user4	26 жовт. 2019 р., 14:30:00	8	26 жовт. 2019 р., 18:10:48	26 жовт. 2019 р., 18:40:52	26 жовт. 2019 р., 21:40:34	26 жовт. 2019 р., 21:16:24	26 жовт. 2019 р., 23:03:22	26 жовт. 2019 р., 23:59:18	Component 5	<a href="#">Знайти ланцюг</a>
Process 21	Yurii	12 трав. 2020 р., 03:00:00	7	-	12 трав. 2020 р., 06:24:49	-	12 трав. 2020 р., 09:08:31	-	12 трав. 2020 р., 09:33:19		<a href="#">Знайти ланцюг</a>
Назва процесу	Відповідальний	Час початку	Очікувана тривалість (годин)	Завершено 50%	Прогноз 50%	Завершено 75%	Прогноз 75%	Дата завершення	Прогноз 100%	Залучені компоненти	

Рисунок 5.8 – Сторінка «Процеси» з виконаним прогнозуванням дати і часу завершення.

Як можна бачити із рисунка 5.8 прогнозування виконано, в тому числі й для процесу «Process 21», який переглядався і раніше.

Як продовження роботи зі сторінкою «Процеси» було натиснуто на кнопку 4 «Знайти ланцюг» для процесу «Process 5», як результат знайдено ланцюг для цього процесу, що зображено на рисунку 5.9.

Аналіз і прогнозування процесів

Домашня

Процеси

user7

Вийти

### Найдовший ланцюг з Process 5 процесу

Назва процесу	Відповідальний	Очікувана тривалість
Process 5	user2	10
Process 6	user4	21
Process 9	user3	8
Process 10	user2	9
Process 12	user5	15
Process 14	user5	12
Process 15	user2	5
Process 17	user3	4
Process 18	user2	5
Process 19	user2	10

Рисунок 5.9 – Сторінка «Найдовший ланцюг з процесу»

Дійсно на рисунку 5.9 можна побачити послідовний список процесів, що утворюють найдовший ланцюг, для якого початком є процес «Process 5». Натиснемо кнопку «Вийти» та авторизуємося під користувачем, що має права адміністратора. Домашню сторінку користувача із правами адміністратора зображено на рисунку 5.10.



Привіт, користувачу!

Це система аналізу та прогнозування інформаційно-комунікаційних процесів

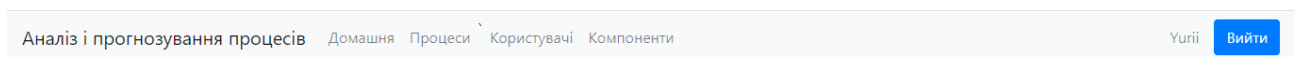
Рисунок 5.10 – Домашня сторінка користувача з правами адміністратора

До нових кнопок, що з'являються для адміністратора, належать такі:

- кнопка 1, «Користувачі» – дозволяє перейти на сторінку з показом усіх користувачів системи;
- кнопка 2, «Компоненти» – дозволяє перейти на сторінку з показом наявних компонентів у системі.

Натиснемо кнопку 1 і перейдемо на сторінку зі списком користувачів.

Її детально зображено на рисунку 5.11.



### Список користувачів

Ім'я	Ролі	Досвід (роки)	1
user1	ADMIN, USER	5	Редагувати
user2	ADMIN	2	Редагувати
user3	USER	3	Редагувати
user4	USER	4	Редагувати
user5	ADMIN	8	Редагувати
Yurii	ADMIN, USER	4	Редагувати
user7	USER	0	Редагувати

Рисунок 5.11 – Сторінка «Список користувачів»

Серед нових можливостей біля кожного наявного користувача доступна кнопка 1 – «Редагувати», що дозволяє вносити зміни щодо користувача. Натиснемо вказану кнопку поруч з користувачем «user7». Сторінка «Редактор користувача» стає доступною. Розглянемо її на рисунку 5.12.

Рисунок 5.12 – Сторінка «Редактор користувача»

На сторінці з рисунка 5.12 можна задати нове ім'я для обраного користувача, змінити його досвід та перелік ролей. Збереження змін доступне за натиском кнопки «Зберегти». Збережемо зміни й перейдемо на сторінку і натиснемо на кнопку «Компоненти». Буде відкрито сторінку з переліком наявних компонентів. Її вигляд представлено на рисунку 5.13.

Назва компонента	Складність
Component 1	LOW
Component 2	MEDIUM
Component 3	LOW
Component 4	HIGH
Component 5	HIGH
Component 6	MEDIUM

Рисунок 5.13 – Сторінка «Компоненти»

На даній сторінці з'являється кнопка 1 – «Додати новий компонент». Вона розгортає форму для додання нового компоненту. Натиснемо на дану кнопку. Як результат можна побачити цю ж сторінку з відкритою формою. Переглянути її можна на рисунку 5.14.

Додати новий компонент

Назва компонента

Component 7A

Складність

HIGH

1

Додати

Назва компонента	Складність
Component 1	LOW
Component 2	MEDIUM
Component 3	LOW
Component 4	HIGH
Component 5	HIGH
Component 6	MEDIUM

Рисунок 5.14 – Сторінка «Компоненти» з можливістю додати новий компонент

На сторінці з рисунка 5.14 з'являється кнопка 1 – «Додати», що дозволяє додати новий компонент із заданою назвою та обраним рівнем складності. Вказавши ці параметри натиснемо на кнопку. Результат взаємодії можна побачити на рисунку 5.15.

Додати новий компонент

Назва компонента	Складність
Component 1	LOW
Component 2	MEDIUM
Component 3	LOW
Component 4	HIGH
Component 5	HIGH
Component 6	MEDIUM
Component 7A	HIGH

Рисунок 5.15 – Сторінка «Компоненти» з доданим новим компонентом

Перейдемо на сторінку «Процеси». Для користувача із правами адміністратора вона має вигляд аналогічний тому, що на рисунку 5.16.

## Зробити прогноз

localhost:8080/addProcess

[Вийти](#)

## 1

58

16		2019 р., 09:00:00		2019 р., 12:44:38	2019 р., 12:06:42	2019 р., 14:15:22	2019 р., 14:10:15	2019 р., 16:46:34	2019 р., 15:58:10		ланцюг
Process 17	user3	26 жовт. 2019 р., 09:00:00	4	26 жовт. 2019 р., 11:25:44	26 жовт. 2019 р., 11:02:45	26 жовт. 2019 р., 13:02:53	26 жовт. 2019 р., 12:24:39	26 жовт. 2019 р., 13:57:32	26 жовт. 2019 р., 13:41:45	Component 5	<a href="#">Знайти ланцюг</a>
Process 18	user2	26 жовт. 2019 р., 10:50:00	5	26 жовт. 2019 р., 13:13:14	26 жовт. 2019 р., 13:16:04	26 жовт. 2019 р., 14:31:41	26 жовт. 2019 р., 15:03:56	26 жовт. 2019 р., 16:13:59	26 жовт. 2019 р., 16:01:05	Component 2	<a href="#">Знайти ланцюг</a>
Process 19	user2	26 жовт. 2019 р., 12:40:00	10	26 жовт. 2019 р., 16:50:34	26 жовт. 2019 р., 17:38:03	26 жовт. 2019 р., 19:39:41	26 жовт. 2019 р., 20:59:26	27 жовт. 2019 р., 00:02:47	26 жовт. 2019 р., 23:01:51	Component 1	<a href="#">Знайти ланцюг</a>
Process 20	user4	26 жовт. 2019 р., 14:30:00	8	26 жовт. 2019 р., 18:10:48	26 жовт. 2019 р., 18:40:52	26 жовт. 2019 р., 21:40:34	26 жовт. 2019 р., 21:16:24	26 жовт. 2019 р., 23:03:22	26 жовт. 2019 р., 23:59:18	Component 5	<a href="#">Знайти ланцюг</a>
Process 21	Yurii	12 трав. 2020 р., 03:00:00	7	-	12 трав. 2020 р., 06:24:49	-	12 трав. 2020 р., 09:08:31	-	12 трав. 2020 р., 09:33:19		<a href="#">Знайти ланцюг</a>
Process 22	Yurii	24 трав. 2020 р., 20:15:00	14	-	-	-	-	-	-	Component 1, Component 6, Component 7A	<a href="#">Знайти ланцюг</a>
Назва процесу	Відповідальний	Час початку	Очікувана тривалість (годин)	Завершено 50%	Прогноз 50%	Завершено 75%	Прогноз 75%	Дата завершення	Прогноз 100%	Залучені компоненти	

Рисунок 5.18 – Сторінка «Процеси» із доданим процесом

Можемо пересвідчитися, що процес «Process 22» наявний і його параметри саме такі, як було задано на рисунку 5.17.

## 5.2 Випробування програмного продукту

Щоби пересвідчитися у правильності роботи розробленого застосунку було проведено тестування програмного продукту.

### 5.2.1 Мета випробувань

Метою випробувань є перевірка відповідності функціоналу розробленої системи аналізу та прогнозування інформаційно-комунікаційних процесів заявленому у технічному завданні.

### 5.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

					ДП 6319.00.000 ПЗ					Арк.
Змн.	Арк.	№ докум.	Підпис	Дата						59

### 5.2.3 Результати випробувань

Розроблений застосунок було перевірено на відповідність вимогам. Опис випробувань та їх результати наведено у таблицях 5.1 – 5.17.

Тести містять перевірку основної функціональності системи – навігація, реєстрація та авторизація, пошук найдовшого ланцюга, прогнозування часу і дат завершення процесів, додавання компонентів та процесів. Сценарії тестування та результати наведено в таблицях 5.1 – 5.17.

Таблиця 5.1 – Тест-кейс 1

Назва тесту	Показ домашньої сторінки
Передумови	1. Запущено веббраузер
Кроки	1. В адресний рядок введено <a href="http://localhost:8080/">http://localhost:8080/</a> 2. Натиснуто кнопку для переходу на сторінку
Очікуваний результат	Показано домашню сторінку застосунку як для неавторизованого користувача
Чи відповідає спостережуваний результат очікуваному?	Так

Таблиця 5.2 – Тест-кейс 2

Назва тесту	Перехід на сторінку авторизації
Передумови	1. Показано домашню сторінку застосунку.
Кроки	1. Натиснуто кнопку «Увійти»
Очікуваний результат	Відображається сторінка авторизації з полями «Ім'я користувача» та «Пароль»
Чи відповідає спостережуваний результат очікуваному?	Так



Таблиця 5.3 – Тест-кейс 3

Назва тесту	Перехід на сторінку реєстрації
Передумови	1. Показано сторінку авторизації.
Кроки	1. Натиснуто кнопку «Зареєструватися?»
Очікуваний результат	Показано сторінку реєстрації з полями «Ім'я користувача» та «Пароль»
Чи відповідає спостережуваний результат очікуваному?	Так

Таблиця 5.4 – Тест-кейс 4

Назва тесту	Реєстрація нового користувача
Передумови	1. Показ сторінки реєстрації
Кроки	1. Введено непорожні значення в комірки полів «Ім'я користувача» і «Пароль» 2. Натиснуто кнопку «Зареєструватися»
Очікуваний результат	Нового користувача додано
Чи відповідає спостережуваний результат очікуваному?	Так

Таблиця 5.5 – Тест-кейс 5

Назва тесту	Авторизація зареєстрованого користувача
Передумови	1. Показ сторінки авторизації
Кроки	1. Введено непорожні значення в комірки полів «Ім'я користувача» і «Пароль», що відповідають наявному в системі користувачеві 2. Натиснуто кнопку «Увійти»

Продовження таблиці 5.5

Очікуваний результат	Авторизацію пройдено, показано домашню сторінку застосунку, в навігаційній панелі показано ім'я користувача.
Чи відповідає спостережуваний результат очікуваному?	Так

Таблиця 5.6 – Тест-кейс 6

Назва тесту	Пошук процесу за назвою
Передумови	1. Користувач авторизований у системі 2. У системі наявні процеси
Кроки	1. Натиснуто кнопку «Процеси» на навігаційній панелі.
Очікуваний результат	Показано сторінку «Процеси», на якій видно таблицю з процесами, кнопки для пошуку процесу за назвою, додавання нового процесу, виконання прогнозу
Чи відповідає спостережуваний результат очікуваному?	Так

Таблиця 5.7 – Тест-кейс 7

Назва тесту	Пошук процесу за назвою
Передумови	1. Показ сторінки «Процеси». 2. Процес із шуканою назвою наявний у системі..
Кроки	1. Задано назву процесу у відповідній комірці 2. Натиснуто кнопку «Шукати»
Очікуваний результат	Показ сторінки «Процеси», на якій видно таблицю зі знайденим процесом.

## Продовження таблиці 5.7

Чи відповідає спостережуваний результат очікуваному?	Так
--	-----

Таблиця 5.8 – Тест-кейс 8

Назва тесту	Виконання прогнозування часу і дати завершення процесу
Передумови	1. Показ сторінки «Процеси». 2. У системі наявні процеси.
Кроки	1. Вказано обсяг навчальної вибірки у відповідному полі. 2. Натиснуто кнопку «Зробити прогноз»
Очікуваний результат	Виконано прогнозування, показано сторінка «Процеси», на якій для усіх процесів, що не потрапили у навчальну вибірку показано прогнозовану дату і час завершення процесу.
Чи відповідає спостережуваний результат очікуваному?	Так

Таблиця 5.9 – Тест-кейс 9

Назва тесту	Пошук найдовшого ланцюга
Передумови	1. Показ сторінки «Процеси» 2. У системі наявні процеси
Кроки	1. Поблизу конкретного процесу натиснуто кнопку «Знайти ланцюг»
Очікуваний результат	Знайдено найдовший ланцюг, що починається в обраному процесі, відображається сторінка «Найдовший ланцюг з процесу»

## Продовження таблиці 5.9

Чи відповідає спостережуваний результат очікуваному?	Так
--	-----

Таблиця 5.10 – Тест-кейс 10

Назва тесту	Авторизація як адміністратора
Передумови	1. Показ сторінки авторизації
Кроки	1. Введено «Ім'я користувача» та «Пароль», що відповідають користувачеві із правами адміністратора
Очікуваний результат	Авторизацію пройдено, показано домашню сторінку застосунку, в навігаційній панелі є ім'я користувача, про те, що користувач – адміністратор, свідчать кнопки «Компоненти» і «Користувачі» на навігаційній панелі.
Чи відповідає спостережуваний результат очікуваному?	Так

Таблиця 5.11 – Тест-кейс 11

Назва тесту	Перегляд сторінки «Компоненти»
Передумови	1. Виконано вхід у систему користувачем із правами адміністратора
Кроки	1. Натиснуто кнопку «Компоненти» на навігаційній панелі
Очікуваний результат	Показано сторінку «Компоненти», на якій показано усі наявні компоненти.
Чи відповідає спостережуваний результат очікуваному?	Так

Таблиця 5.12 – Тест-кейс 12

Назва тесту	Додання нового компонента
Передумови	1. Виконано вхід у систему користувачем із правами адміністратора 2. Показ сторінки «Компоненти»
Кроки	1. Натиснуто кнопку «Додання нового компонента» 2. У відповідних полях введено назву компонента та обрано його складність. 3. Натиснуто кнопку «Додати»
Очікуваний результат	Показано сторінку «Компоненти», у списку з'явився новий компонент.
Чи відповідає спостережуваний результат очікуваному?	Так

Таблиця 5.13 – Тест-кейс 13

Назва тесту	Перегляд сторінки «Користувачі»
Передумови	1. Виконано вхід у систему користувачем із правами адміністратора
Кроки	1. Натиснуто кнопку «Користувачі» на навігаційній панелі.
Очікуваний результат	Показано сторінку «Список користувачів», на якій показано наявні у системі користувачі
Чи відповідає спостережуваний результат очікуваному?	Так

Таблиця 5.14 – Тест-кейс 14

Назва тесту	Редагування користувача
Передумови	<p>1. Виконано вхід у систему користувачем із правами адміністратора</p> <p>2. Показ сторінки «Список користувачів»</p> <p>3. Натиснуто кнопку «Редагувати» поруч конкретним користувачем.</p>
Кроки	<p>1. На сторінці «Редактор користувача» внесено бажані зміни у інформацію щодо користувача.</p> <p>2. Натиснуто кнопку «Зберегти»</p>
Очікуваний результат	Показано сторінку «Список користувачів», на якій показано інформацію про користувачів, в тому числі оновлену про користувача, якого редагували.
Чи відповідає спостережуваний результат очікуваному?	Так

Таблиця 5.15 – Тест-кейс 15

Назва тесту	Перехід на сторінку «Додавання нового процесу»
Передумови	<p>1. Виконано вхід у систему користувачем із правами адміністратора</p> <p>2. Показ сторінки «Процеси»</p>
Кроки	3. Натиснуто кнопку «Додати новий процес»

Продовження таблиці 5.15

Очікуваний результат	Показано сторінку «Список користувачів», на якій наявні у системі користувачі
Чи відповідає спостережуваний результат очікуваному?	Так

Таблиця 5.16 – Тест-кейс 16

Назва тесту	Додавання нового процесу
Передумови	<ol style="list-style-type: none"> <li>Виконано вхід у систему користувачем із правами адміністратора</li> <li>Показ сторінки «Додавання нового процесу»</li> </ol>
Кроки	<ol style="list-style-type: none"> <li>Заповнено поля «Назва процесу», «Дата початку», «Очікувана тривалість».</li> <li>Натиснуто кнопку «Додати процес»</li> </ol>
Очікуваний результат	Показано сторінку «Процеси», в кінці списку наявний процес із щойно введеними параметрами.
Чи відповідає спостережуваний результат очікуваному?	Так

Таблиця 5.17 – Тест-кейс 17

Назва тесту	Вихід з системи
Передумови	1. Користувач авторизований у системі.
Кроки	1. Натиснуто кнопку «Вийти»

## Продовження таблиці 5.17

Очікуваний результат	Показано домашню сторінку системи у вигляді для неавторизованого користувача.
Чи відповідає спостережуваний результат очікуваному?	Так

**Висновок до розділу**

У даному розділі розглянуто функціональність розробленого застосунку, зокрема через представлення екранних форм.

У підрозділі «Керівництво користувача» розглянуто інтерфейс користувача застосунку. Надано опис меню користувача, в залежності від його ролі у системі: неавторизований користувач, користувач чи адміністратор. Розглянуто реєстрацію нового користувача та авторизацію. Переглянуто сторінки прогнозування процесів та пошуку найдовшого ланцюга. Розглянуто можливості додавання нових компонентів та процесів.

У підрозділі «Випробування програмного продукту» сформульовано мету та загальні положення даних випробувань. Докладно наведені тестові сценарії, що перевіряють правильність роботи розробленого застосунку та відповідність заданій в технічному завданні функціональності.



## ЗАГАЛЬНІ ВИСНОВКИ

Пояснювальна записка дипломного проєкту надає детальний опис виконаної роботи щодо розробки системи аналізу та прогнозування інформаційно-комунікаційних процесів.

У розділі «Загальні положення» визначено мету та призначення розробленої системи. Виконано опис предметної області дипломного проєкту. Вказано особливості роботи з процесами, зосереджено увагу на основних складнощах, що можуть виникати.

Розділ «Загальні положення» присвячено опису процесу діяльності і опису функціональної моделі, яка представлена за допомогою схеми варіантів використання, у якій вказано дії, що можуть виконувати користувачі системи. Ця схема є основою функціональних вимог до системи і визначає, що повинна виконувати система. У підрозділі «Огляд наявних аналогів» розглянуто системи, що мають подібний функціонал і водночас вказано їх відмінності від бажаної системи, вказано на їх недоліки для роботи з предметним середовищем.

Розділ «Інформаційне забезпечення» надає опис вхідних та вихідних даних системи. Визначено, що вхідними даними є інформація про процеси, зв'язки між ними, залучені до їх роботи компоненти та користувачів, що за них відповідають. Вихідними даними є результати прогнозування – час і дата завершення 50%, 75% та усього процесу та найдовший ланцюг, що починається в обраному процесі.

Розділ «Математичне забезпечення» містить змістовні та формалізовані постановки задачі прогнозування дати і часу завершення процесів та задачі пошуку найдовшого ланцюга. Запропоновано методи розв'язання, надано їх алгоритми та обґрунтовано доцільність використання названих методів.

Розділ «Програмне та технічне забезпечення» покриває широкий аспект питань пов'язаних і програмною реалізацією системи. У розділі детально описано технології та засоби розробки, що використовувалися для

					ДП 6319.00.000 ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

створення застосунку. Систему розроблено у вигляді вебзастосунку. Він вимагає для своєї роботи наявності технічного та програмного забезпечення, що має бути використовуване. У розділі описано архітектуру розробленого застосунку, вказано пакети, із яких складається продукт, і вказано їх мету. Деякі процеси, що відбуваються у системі, представлено на схемах послідовності, а саме: пошук найдовшого ланцюга і додавання нового процесу. Для пакета «Контролер», що відповідає за безпосередню роботу із користувачем системи, надано опис усіх функцій та їх параметрів.

«Технологічний розділ» присвячено роботі користувача із системою. Керівництво користувача, у якому представлено екранні форми застосунку, допомагає якнайкраще зрозуміти функціональність системи та орієнтуватися в користуванні нею. Також у цьому розділі наведено основні тестові сценарії для перевірки функціональності системи. У результаті проведення тестування визначено, що застосунок працює правильно. Він відповідає усім функціональним вимогам, що наведені у технічному завданні до дипломного проєкту.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Definition of Process by Merriam-Webster [Електронний ресурс] – Режим доступу до ресурсу: <https://www.merriam-webster.com/thesaurus/process>.
2. Parmenter D. Key performance indicators : developing, implementing, and using winning KPIs / David Parmenter. – Hoboken: Wiley, 2015. – 448 с. – (Third edition).
3. Процюк Ю. В. Підхід до аналізу та прогнозування інформаційно-комунікаційних процесів / Ю. В. Процюк, Е. В. Жаріков. // Матеріали четвертої всеукраїнської науково-практичної конференції молодих вчених та студентів "Інформаційні системи та технології управління". – 2020. – С. 98–101.
4. Burgin M. Theory of Knowledge Structures and Processes / Mark Burgin. – Singapore: World Scientific Publishing Co., 2017. – 948 с. – (World Scientific Series in Information Studies).
5. Krill P. Why R? The pros and cons of the R language [Електронний ресурс] / Paul Krill. – 2015. – Режим доступу до ресурсу: <https://www.infoworld.com/article/2940864/r-programming-language-statistical-data-analysis.html>.
6. Maidamisa A. Project Management using Critical Path Method (CPM) / Ahmad Maidamisa. // Global Journal of Pure and Applied Sciences. – 2013. – №18. – С. 197–206.
7. Levy F. K. The ABCs of the Critical Path Method [Електронний ресурс] / F. K. Levy, G. L. Thompson, J. D. Wiest // Harvard Business Review. – 1963. – Режим доступу до ресурсу: <https://hbr.org/1963/09/the-abcs-of-the-critical-path-method>.
8. Caudill M. Neural networks primer, part I / Maureen Caudill. // AI Expert. – 1987.
9. Aleksander I. An introduction to neural computing / I. Aleksander, H. Morton. – London: International Thomson Computer Press, 1995. – 284 с.

					ДП 6319.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		71

- 10.Rojas R. Neural Networks - A Systematic Introduction / Raul Rojas. – Berlin, New-York: Springer-Verlag, 1996. – 502 с.
- 11.Хайкин С. Нейронные сети: полный курс / Саймон Хайкин. – Москва: Издательский дом "Вильямс", 2006. – 1104 с.
- 12.Arnold K. The Java Programming Language, Fourth Edition / K. Arnold, J. Gosling, D. Holmes. – Boston: Addison Wesley Professional, 2005. – 928 с.
- 13.Walls C. Spring in Action, Fifth Edition / Craig Walls. – New York: Manning Publications, 2018. – 520 с.
- 14.Mulders M. What Is Spring Boot? [Электронный ресурс] / Michiel Mulders. – 2019. – Режим доступа до ресурсу: <https://stackify.com/what-is-spring-boot/>.
- 15.Gupta A. Pros and Cons of Bootstrap & Foundation – Know what you need! [Электронный ресурс] / Aayush Gupta. – 2017. – Режим доступа до ресурсу: <https://vmokshagroup.com/blog/bootstrap-advantages/>.
- 16.Schwartz B. High Performance MySQL, Third Edition / B. Schwartz, P. Zaitsev, V. Tkachenko. – Sebastopol: O'Reilly, 2012. – 826 с.
- 17.Branson T. 8 Major Advantages of Using MySQL [Электронный ресурс] / Tony Branson. – 2016. – Режим доступа до ресурсу: <https://www.datamation.com/storage/8-major-advantages-of-using-mysql.html>.
- 18.Bauer C. Java Persistence with Hibernate 2nd Edition / C. Bauer, G. King, G. Gregory. – New York: Manning Publications, 2015. – 608 с.
- 19.Pop D. Designing an MVC Model for Rapid Web Application Development / D. Pop, A. Altar. // Procedia Engineering. – 2014. – №69. – С. 1172–1179.

Додаток А

**Тексти програмного коду**

**Система аналізу та прогнозування інформаційно-комунікаційних  
процесів**

(Найменування програми (документа))

*DVD-R*

(Вид носія даних)

*12 арк, 415 Кб*

(Обсяг програми (документа) , арк.,) Кб)

Київ – 2020 року

					ДП 6319.00.000 ПЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		

```
package com.example.PF.controller;

import com.example.PF.domain.Complexity;
import com.example.PF.domain.Component;
import com.example.PF.domain.Proc;
import com.example.PF.repos.ComponentRepo;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;

@Controller
public class ComponentController {
    @Autowired
    ComponentRepo componentRepo;
    @GetMapping("/components")
    public String allComponents(Model model){
        Iterable<Component> components = componentRepo.findAll();
        model.addAttribute("components", components);
        return "components";
    }
    @PostMapping("/components")
    public String addComponent(@RequestParam String name, @RequestParam String
complexity){
        Complexity complexity1 = Complexity.valueOf(complexity);
        Component component = new Component(name,complexity1);
        componentRepo.save(component);
        return "redirect:/components";
    }
}
```

```
package com.example.PF.controller;

import com.example.PF.CP.FindPath;
import com.example.PF.NN.Calculation;
import com.example.PF.domain.Connection;
import com.example.PF.domain.Proc;
import com.example.PF.repos.ConnectionRepo;
import com.example.PF.repos.ProcRepo;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
```

					ДП 6319.00.000 ПЗ	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дата		

```

import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;

@Controller
public class MainController {
    @Autowired
    private ProcRepo procRepo;
    @Autowired
    ConnectionRepo connectionRepo;
    @Value("${upload.path}")
    private String uploadPath;

    @GetMapping("/")
    public String greeting() {
        return "index";
    }

    @GetMapping("main")
    public String main(@RequestParam(required = false, defaultValue = "") String filter, Model
model) {
        Iterable<Proc> procs = procRepo.findAll();

        if (filter != null && !filter.isEmpty()) {
            procs = procRepo.findByName(filter);
        } else {
            procs = procRepo.findAll();
        }

        model.addAttribute("procs", procs);
        model.addAttribute("filter", filter);

        return "main";
    }

    @GetMapping("main/{proc}")
    public String findPath(@PathVariable Proc proc, Model model) throws IOException {
        model.addAttribute("proc", proc);
        Iterable<Connection> connections = connectionRepo.findAll();
        List<Connection> connections1 = new ArrayList<>();
        connections.forEach(connections1::add);
        //ist<Connection> connections =
entityManager.createNamedQuery("getNeedConnection").setParameter(1,
proc.getId().intValue()).getResultList();
        List<Integer> pathFind = FindPath.getPath(connections1, proc);
        List<Proc> procs = new ArrayList<>();
        List<Long> findPath2 = new ArrayList<>();

```

```

    if (pathFind == null) {
        procs.add(proc);
    } else {
        for (Integer i : pathFind) {
            findPath2.add(new Long(i));
        }
        for (Long num : findPath2) {
            procRepo.findById(num).ifPresent(value -> procs.add(value));
        }
    }
    model.addAttribute("procs", procs);
    return "findPath";
}

@PostMapping("main")
public String makePrediction(@RequestParam(required = false, defaultValue = "") String
filter,
                             @RequestParam(required = true, defaultValue = "10") Integer trainSize,
                             Model model) throws Exception {
    Iterable<Proc> procs = procRepo.findAll();
    List<Proc> procs1 = new ArrayList<>();
    procs.forEach(procs1::add);
    int size = procs1.size();
    if (trainSize > size || trainSize < 2) {
        trainSize = (int) (size) / 2;
    }
    List<Proc> first = new ArrayList<>(procs1.subList(0, trainSize));
    List<Proc> second = new ArrayList<>(procs1.subList(trainSize, size));
    Iterable<Proc> getProcs = Calculation.predictDates(first, second);
    for (Proc proc : getProcs) {
        procRepo.save(proc);
    }
    model.addAttribute("procs", procs);
    model.addAttribute("filter", filter);
    return "redirect:/main";
}
}

```

```
package com.example.PF.controller;
```

```

import com.example.PF.domain.Component;
import com.example.PF.domain.Proc;
import com.example.PF.domain.User;
import com.example.PF.repos.ComponentRepo;
import com.example.PF.repos.ProcRepo;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.format.annotation.DateTimeFormat;
import org.springframework.security.core.annotation.AuthenticationPrincipal;
import org.springframework.stereotype.Controller;

```



```

import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;

import java.util.*;

@Controller
public class ProcessController {
    @Autowired
    ProcRepo procRepo;
    @Autowired
    ComponentRepo componentRepo;

    @GetMapping("/addProcess")
    public String addForm(Model model) {

        Iterable<Component> components = componentRepo.findAll();
        model.addAttribute("components", components);
        return "addProcess";
    }

    @PostMapping("/addProcess")
    public String addProcess(
        @AuthenticationPrincipal User user,
        @RequestParam String name,
        @RequestParam("startDate") @DateTimeFormat(iso = DateTimeFormat.ISO.DATE)
        Date startDate,
        @RequestParam Integer expectedDuration,
        @RequestParam Long[] compids) {
        List<Long> listCompids = Arrays.asList(compids);

        Set<Component> componentsAdd = new HashSet<>();
        Proc proc = new Proc(name, startDate, expectedDuration, user);
        for (Long id : listCompids) {
            Component component = componentRepo.findById(id).orElse(null);
            componentsAdd.add(component);
        }
        proc.setComponents(componentsAdd);
        procRepo.save(proc);
        return "redirect:/main";
    }
}

package com.example.PF.controller;

import com.example.PF.domain.Role;
import com.example.PF.domain.User;

```

					ДП 6319.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		77

```

import com.example.PF.repos.UserRepo;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;

import java.util.Collections;
import java.util.Map;

@Controller
public class RegistrationController {
    @Autowired
    private UserRepo userRepo;

    @GetMapping("/login")
    public String login() {
        return "login";
    }

    @GetMapping("/registration")
    public String registration() {
        return "registration";
    }

    @PostMapping("/registration")
    public String addUser(User user, Map<String, Object> model) {
        User userFromDb = userRepo.findByUsername(user.getUsername());

        if (userFromDb != null) {
            model.put("procs", "User exists!");
            return "registration";
        }
        user.setWorkexperience(0);
        user.setActive(true);
        user.setRoles(Collections.singleton(Role.USER));
        userRepo.save(user);

        return "redirect:/login";
    }
}

```

```
package com.example.PF.controller;
```

```

import com.example.PF.domain.Role;
import com.example.PF.domain.User;
import com.example.PF.repos.UserRepo;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.access.prepost.PreAuthorize;
import org.springframework.stereotype.Controller;

```

					ДП 6319.00.000 ПЗ	Арк.
						78
Змн.	Арк.	№ докум.	Підпис	Дата		

```

import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;

import java.util.Arrays;
import java.util.Map;
import java.util.Set;
import java.util.stream.Collectors;

@Controller
@RequestMapping("/user")
@PreAuthorize("hasAuthority('ADMIN')")
public class UserController {
    @Autowired
    private UserRepo userRepo;

    @GetMapping
    public String userList(Model model) {
        model.addAttribute("users", userRepo.findAll());
        return "userList";
    }

    @GetMapping("/{user}")
    public String userEditForm(@PathVariable User user, Model model) {
        model.addAttribute("user", user);
        model.addAttribute("roles", Role.values());
        return "userEdit";
    }

    @PostMapping
    public String userSave(
        @RequestParam String username,
        @RequestParam Integer workexperience,
        @RequestParam Map<String, String> form,
        @RequestParam("userId") User user
    ) {
        user.setUsername(username);
        user.setWorkexperience(workexperience);
        Set<String> roles = Arrays.stream(Role.values())
            .map(Role::name)
            .collect(Collectors.toSet());

        user.getRoles().clear();

        for (String key : form.keySet()) {
            if (roles.contains(key)) {
                user.getRoles().add(Role.valueOf(key));
            }
        }

        userRepo.save(user);
    }
}

```

					ДП 6319.00.000 ПЗ	Арк.
						79
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return "redirect:/user";
    }
}

package com.example.PF.repos;

import com.example.PF.domain.Component;
import org.springframework.data.repository.CrudRepository;

import java.util.Optional;

public interface ComponentRepo extends CrudRepository<Component, Long> {
    Component findByName(String name);
    Optional<Component> findById (Long id);
}

package com.example.PF.repos;

import com.example.PF.domain.Connection;
import org.springframework.data.repository.CrudRepository;

public interface ConnectionRepo extends CrudRepository<Connection, Long> {
}

package com.example.PF.repos;

import com.example.PF.domain.Proc;
import org.springframework.data.repository.CrudRepository;

import javax.persistence.Id;
import java.util.List;

public interface ProcRepo extends CrudRepository<Proc, Long> {
    List<Proc> findByName(String name);
    Proc findById(Id id);
}

package com.example.PF.repos;

import com.example.PF.domain.User;
import org.springframework.data.jpa.repository.JpaRepository;

public interface UserRepo extends JpaRepository<User, Long> {
    User findByUsername(String username);
}

```

```

package com.example.PF.CP;

import java.io.IOException;
import java.util.LinkedList;

public class ProcessGraphTools {
    //Перевірка графа на правильність побудови
    public static boolean isValid(ProcessGraph graph) {
        Step first = graph.getFirst();
        Step last = graph.getLast();
        if (first.getOutEdges().size() < 1) {
            return false;
        }
        if (first.getInEdges().size() > 1) {
            return false;
        }
        if (last.getInEdges().size() < 1) {
            return false;
        }
        if (last.getOutEdges().size() > 1) {
            return false;
        }
        if (!ProcessGraphTools.isConnected(graph)) {
            return false;
        }
        if (!ProcessGraphTools.isAcyclic(graph)) {
            return false;
        }
        return true;
    }

    //Перевірка графа на зв'язність
    public static boolean isConnected(ProcessGraph graph) {
        LinkedList<Step> steps = graph.getSteps();
        Step first = graph.getFirst();
        Step last = graph.getLast();
        for (Step step : steps) {
            if (!step.equals(first) && !step.equals(last)) {
                if (step.getInEdges().size() == 0 || step.getOutEdges().size() == 0) {
                    return false;
                }
            }
        }
        return true;
    }

    //Перевірка графа на відсутність циклів
    public static boolean isAcyclic(ProcessGraph graph) {
        LinkedList<Step> copySteps = new LinkedList<Step>();
        for (Step step : graph.getSteps()) {
            copySteps.add(new Step(step));
        }
    }
}

```

```

    }
    LinkedList<Edge> copyEdges = new LinkedList<Edge>();
    for (Edge edge : graph.getEdges()) {
        copyEdges.add(new Edge(edge));
    }
    LinkedList<Step> sorted = new LinkedList<Step>();
    LinkedList<Step> stepsNotInEdges = new LinkedList<Step>();
    for (Step step : copySteps) {
        if (step.getInEdges().size() == 0) {
            stepsNotInEdges.add(step);
        }
    }
    while (!stepsNotInEdges.isEmpty()) {
        Step step = stepsNotInEdges.removeFirst();
        sorted.add(step);
        if (!step.getOutEdges().isEmpty()) {
            for (int i = step.getOutEdges().size() - 1; i >= 0; i--) {
                Edge edge = step.getOutEdges().get(i);
                Step compare = edge.getTo();
                int index = copySteps.indexOf(compare);
                Step m = copySteps.get(index);
                copyEdges.remove(edge);
                m.deleteEdge(edge);
                if (m.getInEdges().isEmpty()) {
                    if (!stepsNotInEdges.contains(m)) {
                        stepsNotInEdges.add(m);
                    }
                }
            }
        }
    }
    boolean cycle = false;
    for (Step step : copySteps) {
        if (!step.getInEdges().isEmpty()) {
            cycle = true;
            break;
        }
    }

    return !cycle;
}

//Пошук шляху та знаходження його тривалості
public static LinkedList<Edge> pathWeight(ProcessGraph graph) throws IOException {
    int totalWeight = 0;
    setEarlyLate(graph);
    Step first = graph.getFirst();
    LinkedList<Edge> path = new LinkedList<Edge>();
    LinkedList<Edge> outEdges = first.getOutEdges();
    LinkedList<Step> possibleNextStep = new LinkedList<Step>();
    for (Edge edge : outEdges) {

```

```

        if (!possibleNextStep.contains(edge.getTo()))
            possibleNextStep.add(edge.getTo());
    }
    boolean endStepFound = false;
    while (!endStepFound) {
        for (Step step : possibleNextStep) {
            for (Edge edge : step.getInEdges()) {
                if ((edge.getTo().getEarly() == edge.getTo().getLate())
                    && (edge.getFrom().getEarly() == edge.getFrom().getLate())
                    && (step.getLate() == edge.getCost() + edge.getFrom().getLate())) {
                    path.add(edge);
                    totalWeight += edge.getCost();
                    if (step.getId() == Integer.MAX_VALUE) {
                        endStepFound = true;
                        break;
                    }
                }
                outEdges = step.getOutEdges();
                possibleNextStep = new LinkedList<Step>();
                for (Edge edge1 : outEdges) {
                    if (!possibleNextStep.contains(edge1.getTo()))
                        possibleNextStep.add(edge1.getTo());
                }
                if (step.getId() == Integer.MAX_VALUE)
                    endStepFound = true;
            }
        }
    }
    return path;
}

```

```

public static void setEarlyLate(ProcessGraph graph) {
    if (!ProcessGraphTools.isValid(graph)) {
        throw new IllegalStateException("Graph is invalid");
    }
    setEarly(graph);
    setLate(graph);
}

```

//Встановлення ранн

```

private static void setEarly(ProcessGraph graph) {
    Step first = graph.getFirst();
    first.setEarly(0);
    LinkedList<Edge> outEdges = first.getOutEdges();
    LinkedList<Step> currentSteps = new LinkedList<Step>();
    for (Edge edge : outEdges) {
        if (!currentSteps.contains(edge.getTo()))
            currentSteps.add(edge.getTo());
    }
}

```

Змн.	Арк.	№ докум.	Підпис	Дата

```

while (!currentSteps.isEmpty()) {
    for (int i = currentSteps.size() - 1; i >= 0; i--) {
        int finalEarly = -1;
        int tempEarly = -1;
        Step step = currentSteps.get(i);
        LinkedList<Edge> inEdges = step.getInEdges();
        for (Edge edge : inEdges) {
            if ((edge.getFrom()).getEarly() != -1) {
                tempEarly = (edge.getFrom()).getEarly() + edge.getCost();
            } else {
                break;
            }
            if (tempEarly > finalEarly)
                finalEarly = tempEarly;
            if (step.getEarly() < finalEarly) {
                step.setEarly(finalEarly);
            }
            currentSteps.remove(step);
            LinkedList<Edge> outTempEdges = step.getOutEdges();
            for (Edge edgeTemp : outTempEdges) {
                if (!currentSteps.contains(edgeTemp.getTo())) {
                    currentSteps.add(edgeTemp.getTo());
                }
            }
        }
    }
}

```

```

//Встановлення пізнього початку
private static void setLate(ProcessGraph graph) {
    Step first = graph.getFirst();
    first.setLate(0);
    Step last = graph.getLast();
    for (Edge edge : last.getInEdges()) {
        int late = (edge.getFrom()).getEarly() + edge.getCost();
        if (late > last.getLate()) {
            last.setLate(late);
        }
    }
    LinkedList<Edge> inEdges = last.getInEdges();
    LinkedList<Step> currentSteps = new LinkedList<Step>();
    for (Edge edge : inEdges) {
        if (!currentSteps.contains(edge.getFrom()))
            currentSteps.add(edge.getFrom());
    }
    while (!currentSteps.isEmpty()) {
        for (int i = currentSteps.size() - 1; i >= 0; i--) {
            int finalLate = Integer.MAX_VALUE;
            int tempLate = Integer.MAX_VALUE;

```



```

Step step = currentSteps.get(i);
LinkedList<Edge> retiring = step.getOutEdges();
for (Edge edge : retiring) {
    if ((edge.getTo()).getLate() != Integer.MAX_VALUE) {
        tempLate = (edge.getTo()).getLate() - edge.getCost();
    } else {
        break;
    }
    if (tempLate < finalLate)
        finalLate = tempLate;
    if (step.getLate() > finalLate) {
        step.setLate(finalLate);
    }
    currentSteps.remove(step);
    LinkedList<Edge> inEdgesTemp = step.getInEdges();
    for (Edge edgeTemp : inEdgesTemp) {
        if (!currentSteps.contains(edgeTemp.getFrom())) {
            currentSteps.add(edgeTemp.getFrom());
        }
    }
}
}
}
}
}
}
}

```



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО”  
Кафедра автоматизованих систем обробки інформації і управління

**УЗГОДЖЕНО**

**Керівник проєкту**

\_\_\_\_\_ Едуард ЖАРІКОВ  
(підпис) (вл. ім'я, прізвище)

“13” квітня 2020 р.

**ЗАТВЕРДЖУЮ**

**В.о. завідувача кафедри**

\_\_\_\_\_ Олександр ПАВЛОВ  
(підпис) (вл. ім'я, прізвище)

“14” квітня 2020 р.

Система аналізу та прогнозування інформаційно-комунікаційних  
процесів

**ТЕХНІЧНЕ ЗАВДАННЯ**

Шифр *ДП 6319.01.000 ТЗ*

на 10 сторінках

Київ – 2020 року

# ЗМІСТ

<b>1</b>	<b>ЗАГАЛЬНІ ПОЛОЖЕННЯ .....</b>	<b>3</b>
1.1	ПОВНЕ НАЙМЕНУВАННЯ СИСТЕМИ ТА ЇЇ УМОВНЕ ПОЗНАЧЕННЯ .....	3
1.2	НАЙМЕНУВАННЯ ОРГАНІЗАЦІЇ-ЗАМОВНИКА ТА ОРГАНІЗАЦІЙ-УЧАСНИКІВ РОБІТ.....	3
1.3	ПЕРЕЛІК ДОКУМЕНТІВ, НА ПІДСТАВІ ЯКИХ СТВОРЮЄТЬСЯ СИСТЕМА .....	3
1.4	ПЛАНОВІ ТЕРМІНИ ПОЧАТКУ І ЗАКІНЧЕННЯ РОБОТИ ЗІ СТВОРЕННЯ СИСТЕМИ.....	4
<b>2</b>	<b>ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ .....</b>	<b>5</b>
2.1	ПРИЗНАЧЕННЯ СИСТЕМИ .....	5
2.2	ЦІЛІ СТВОРЕННЯ СИСТЕМИ.....	5
<b>3</b>	<b>ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ.....</b>	<b>6</b>
<b>4</b>	<b>ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....</b>	<b>7</b>
4.1	ВИМОГИ ДО ФУНКЦІОНАЛЬНИХ ХАРАКТЕРИСТИК.....	7
4.2	ВИМОГИ ДО НАДІЙНОСТІ.....	7
4.3	ВИМОГИ ДО СКЛАДУ І ПАРАМЕТРІВ ТЕХНІЧНИХ ЗАСОБІВ .....	8
<b>5</b>	<b>СТАДІЇ І ЕТАПИ РОЗРОБКИ.....</b>	<b>9</b>
<b>6</b>	<b>ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ.....</b>	<b>10</b>
6.1	ВИДИ ВИПРОБУВАНЬ.....	10

					ДП 6319.01.000 ТЗ						
		Прізвище	Підпис	Дата							
Розроб.	Процюк Ю.В..				Система аналізу та прогнозування інформаційно-комунікаційних процесів			Літ.	Лист	Листів	
Перевірив.	Жаріков Е.В.								2	10	
								КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-63			
Н. кон.	Проскура С.Л.										
Затв.	Павлов О.А.										

# 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

## 1.1 Повне найменування системи та її умовне позначення

Повна назва системи: «Система аналізу та прогнозування інформаційно-комунікаційних процесів».

Коротка назва – «Система».

## 1.2 Найменування організації-замовника та організацій-учасників робіт

Замовником проекту є кафедра Автоматизованих систем обробки інформації та управління факультету інформатики та обчислювальної техніки Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Представник замовника: доцент кафедри АСОІУ Жаріков Едуард В'ячеславович. Адреса замовника: м. Київ, п-кт Перемоги 37.

Розробником системи є студент групи ІС-63 кафедри Автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» Процюк Юрій Володимирович.

## 1.3 Перелік документів, на підставі яких створюється система

При розробці системи і створенні проєктно-експлуатаційної документації Виконавець повинен керуватися вимогами наступних нормативних документів:

- ДСТУ 19.201-78. Технічне завдання. Вимоги до змісту і оформлення;
- ДСТУ 34.601-90. Комплекс стандартів на автоматизовані системи.

Автоматизовані системи. Стадії створення;

					ДП 6319.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

- ДСТУ 34.201-89. Інформаційні технології. Комплекс стандартів на автоматизовані системи. Види, комплексність і позначення документів при створенні автоматизованих систем.

#### **1.4 Планові терміни початку і закінчення роботи зі створення системи**

Плановий термін початку робіт по створенню Системи – 17.05.2020.

Плановий термін завершення робіт по створенню Системи – 01.06.2020.

					ДП 6319.01.000 ТЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ

### 2.1 Призначення системи

Призначенням системи є аналіз процесів та прогнозування стану їх виконання.

### 2.2 Цілі створення системи

Метою створення даної системи є покращення організації діяльності з аналізу та прогнозування інформаційно-комунікаційних процесів.

Для досягнення поставленої мети необхідно розв'язати такі задачі:

- провести аналіз процесів, з визначенням параметрів, що найбільше впливають на процеси;
- на основі проведеного аналізу процесів, визначення методів для прогнозування стану виконання процесів;
- розробити програмний застосунок, що здійснює прогнозування плинусу процесів;
- підтримка можливості збереження результату прогнозування;
- провести тестування й оцінку продуктивності розробленого застосунку.

Система може застосовуватись як і окремо, так і вбудовуватися в іншу систему.

					ДП 6319.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

### 3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ

Працювати з застосунком можуть усі користувачі, що матимуть доступ до нього. З допомогою застосунку користувач матиме змогу проаналізувати процеси, побачити результати прогнозування їх плину, щоб використовувати цю інформацію для прийняття в майбутньому управлінських рішень.

Об'єктом автоматизації є діяльність з аналізу та прогнозування інформаційно-комунікаційних процесів.

					ДП 6319.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6



## 4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Вимоги до функціональних характеристик

Застосунок повинен задовольняти потреби його користувачів.  
Застосунок повинен виконувати такі функції:

- а) система надає користувачеві змогу зареєструватися;
- б) система надає користувачеві змогу виконати авторизацію у систему;
- в) система надає користувачеві змогу шукати найдовший ланцюг з процесу;
  - 1) система надає користувачеві змогу обирати процес, для якого шукати найдовший ланцюг;
  - 2) система надає користувачеві змогу переглядати результати пошуку найдовшого ланцюга;
- г) система надає користувачеві змогу прогнозувати стан виконання процесів;
  - 1) система надає користувачеві змогу обирати процеси для прогнозу;
  - 2) система надає користувачеві змогу переглянути результат прогнозу;
  - 3) система зберігає результати виконаного прогнозу;
- д) система надає користувачеві з правами адміністратора змогу виконувати редагування інших користувачів;
- е) система надає користувачеві з правами адміністратора змогу додавати компоненти.

### 4.2 Вимоги до надійності

Програма повинна продовжувати працювати і відновлювати увесь спектр своїх функцій, при виникненні таких позапланових ситуацій:

- при помилках в роботі апаратних засобі (окрім носіїв даних і програм).

					ДП 6319.01.000 ТЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

### 4.3 Вимоги до складу і параметрів технічних засобів

Для правильної роботи розробленого застосунку до складу технічних засобів мають входити:

а) комп'ютер із такою конфігурацією:

- 1) процесор з тактовою частотою не нижче 1.6 ГГц;
- 2) достатній об'єм оперативної пам'яті (не менше 1 ГБ);
- 3) інші складові можуть мати будь-які параметри, оскільки вони впливають на роботу програми неістотно;

б) також має бути встановлене таке програмне забезпечення:

- 1) операційна система Windows 7 або вище;
- 2) СУБД MySQL 8.0 або вище;
- 3) .NET Framework 4.4 або вище;
- 4) JDK 8 або вище;

в) комп'ютерна периферія, а саме:

- 1) монітор;
- 2) маніпулятор-миша;
- 3) клавіатура.

## 5 СТАДІЇ І ЕТАПИ РОЗРОБКИ

№ п/п	Назва етапу роботи	Термін виконання етапу	Результат виконання
1.	Підготовка технічного завдання на розробку програмного продукту	25.04.2020	
2.	Розробка сценарію роботи	30.04.2020	
3.	Розробка інформаційного забезпечення	03.05.2020	
4.	Розробка математичної частини	07.05.2020	
5.	Розробка програмного забезпечення	10.05.2020	
6.	Налагодження програми	13.05.2020	
7.	Тестування програми	20.05.2020	
8.	Фінальне оформлення документації	31.05.2020	
9.	Демонстрація готового застосунку	31.05.2020	

## 6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

Приймання програмного продукту має проводитись не пізніше, ніж 01.06.2020. Перелік необхідних матеріалів, що мають бути надані під час приймання:

- технічне завдання;
- програмний продукт;
- керівництво користувача.

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603-92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

### 6.1 Види випробувань

Для перевірки коректності роботи застосунку буде проведено функціональне тестування. В ході тестування буде проведено випробування основних функціональних характеристик програмного продукту та всієї моделі загалом.

Всі випробування, що виконувалися для перевірки застосунку, мають бути у пояснювальній записці до дипломного проєкту.

					ДП 6319.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10



Власник документу:  
Попенко Володимир Дмитрович

ID перевірки:  
1003881986

Дата перевірки:  
08.06.2020 18:22:20 EEST

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
09.06.2020 16:37:38 EEST

ID користувача:  
77149

Назва документу: Procjyk\_bachelor\_is63

ID файлу: 1003896792 Кількість сторінок: 65 Кількість слів: 9548 Кількість символів: 70791 Розмір файлу: 1,016.58 KB

## 6.06% Схожість

Найбільша схожість: 2.92% з джерело бібліотеки. ID файлу: 5824670

2.83% Схожість з Інтернет джерелами 24 ..... Page 67

5.94% Текстові збіги по Бібліотеці акаунту 202 ..... Page 67

## 0.19% Цитат

Цитати 2 ..... Page 68

Вилучення переліку посилань вимкнено

## 0% Вилучень

Вилучений текст відсутній

## Підміна символів

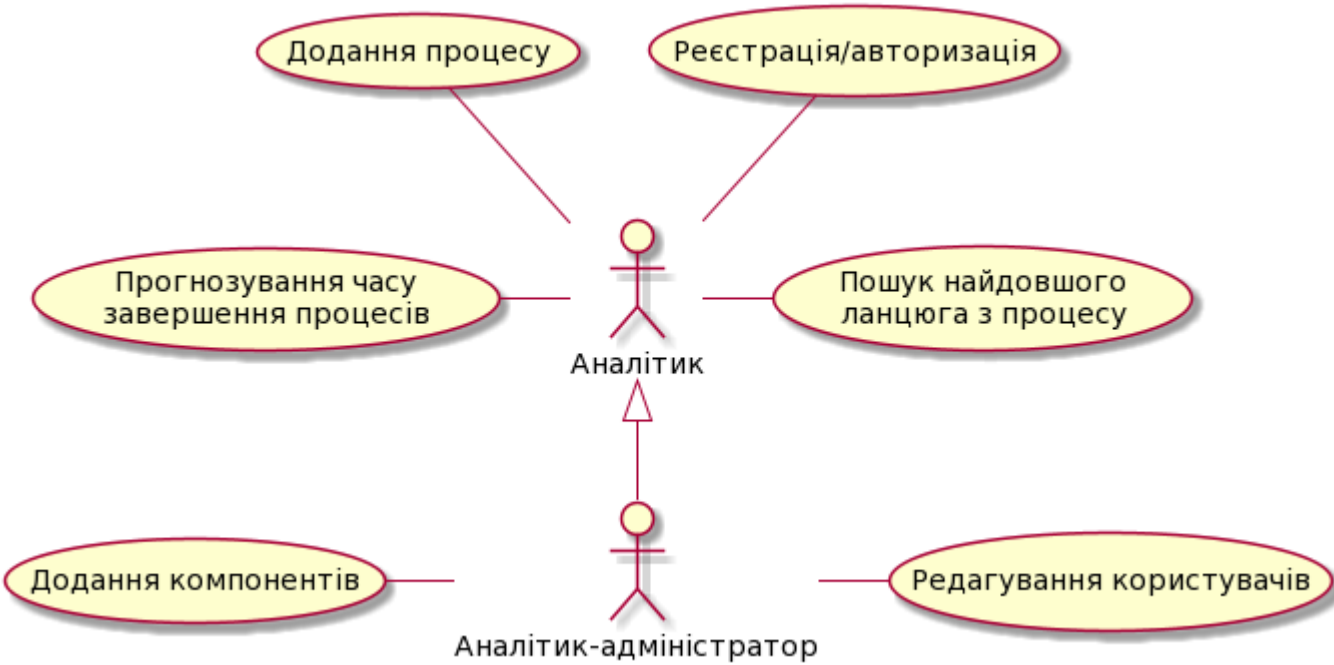
Заміна символів 10

# **Графічний матеріал до дипломного проєкту**

на тему: Система аналізу та прогнозування інформаційно-комунікаційних процесів

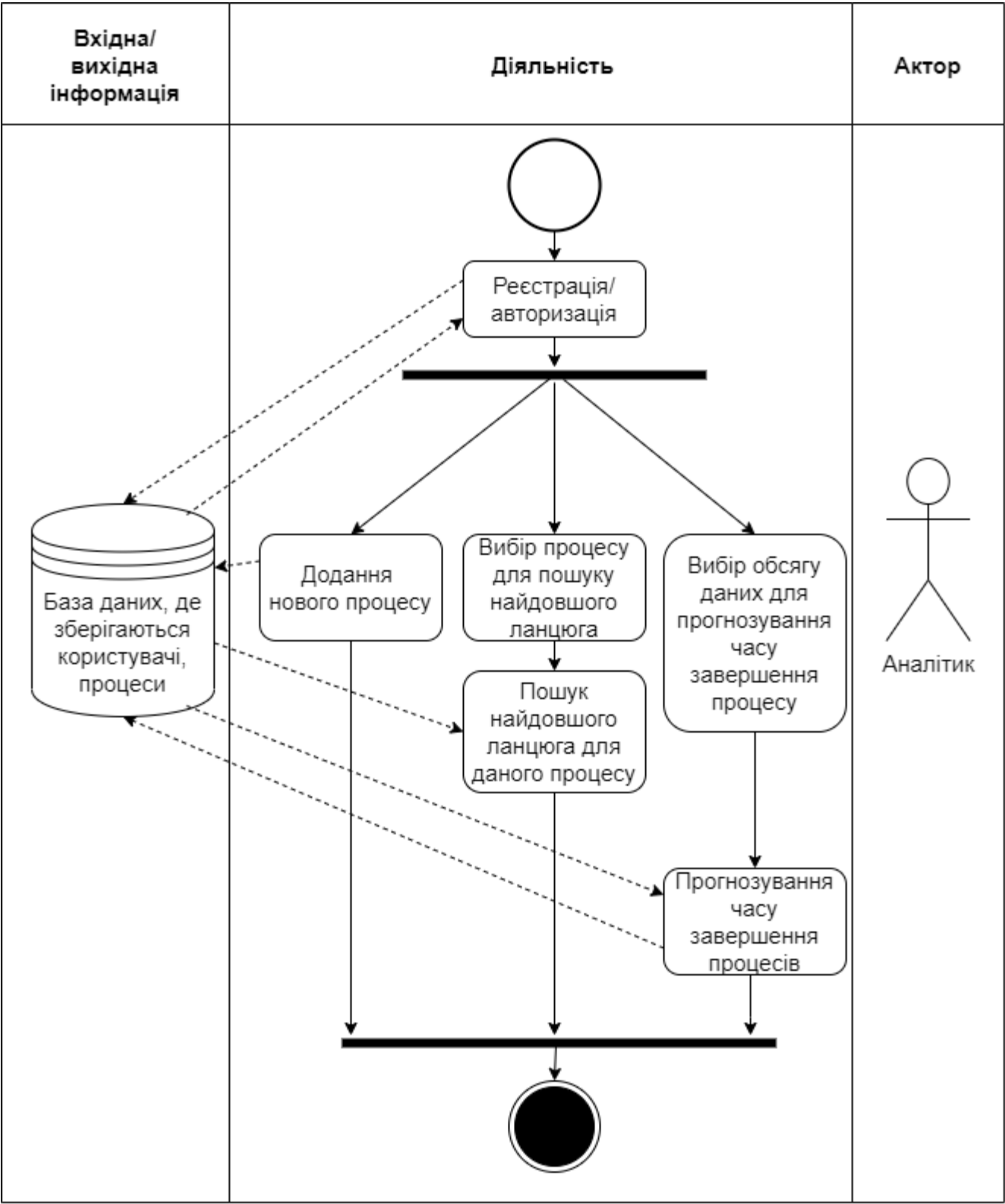
---

Київ – 2020 року

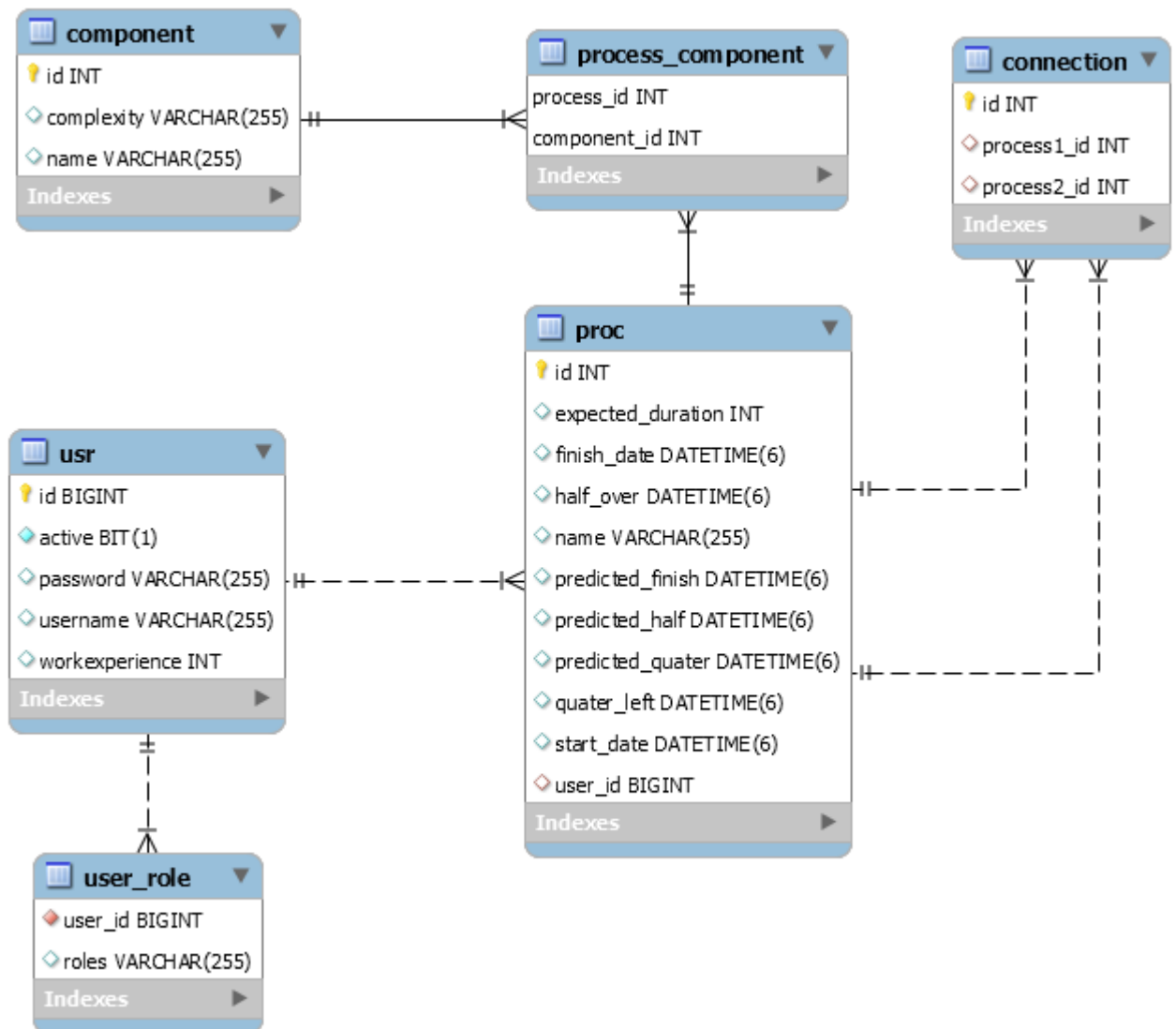


					ДП 6319.02.000 ССВ				
					Схема структурна варіантів використань				
Зм.	Арк.	№ документа	Підпис	Дата					
Розробив		Процюк Ю.В.							
Перевірів		Жаріков Е.В.							
Т. кон.					Система аналізу та прогнозування інформаційно-комунікаційних процесів				
Н. кон.		Проскура С.Л.							
Затвердив		Жаріков Е.В.							
					ЛітераМасаМасштаб				
					Аркуш 1Аркушів 1				
					КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-63				

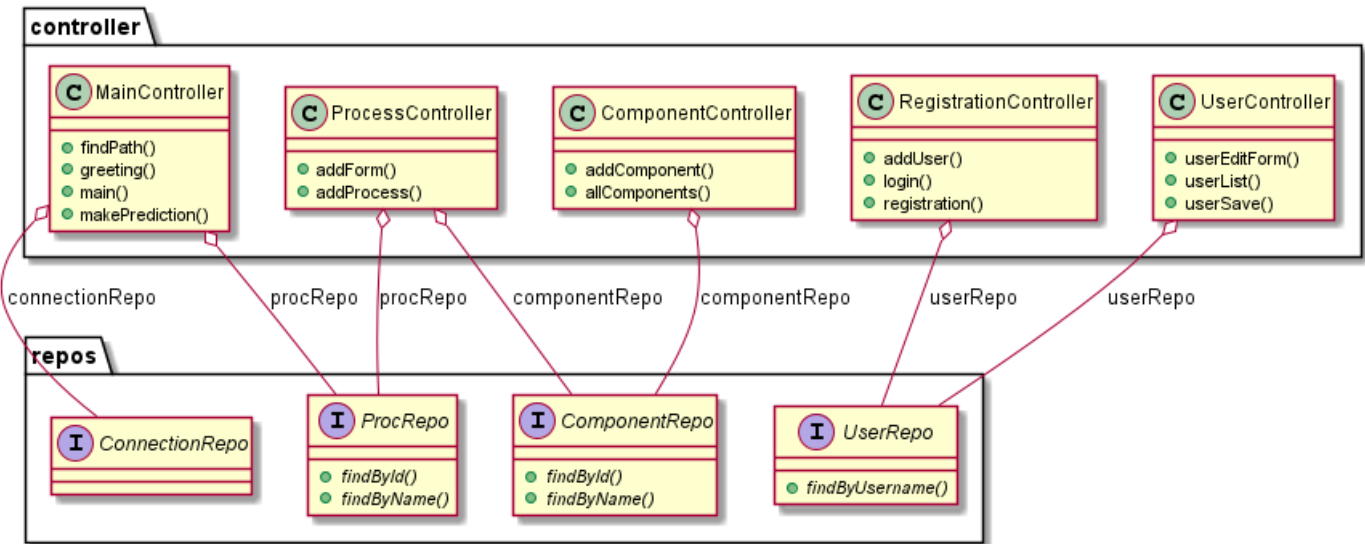




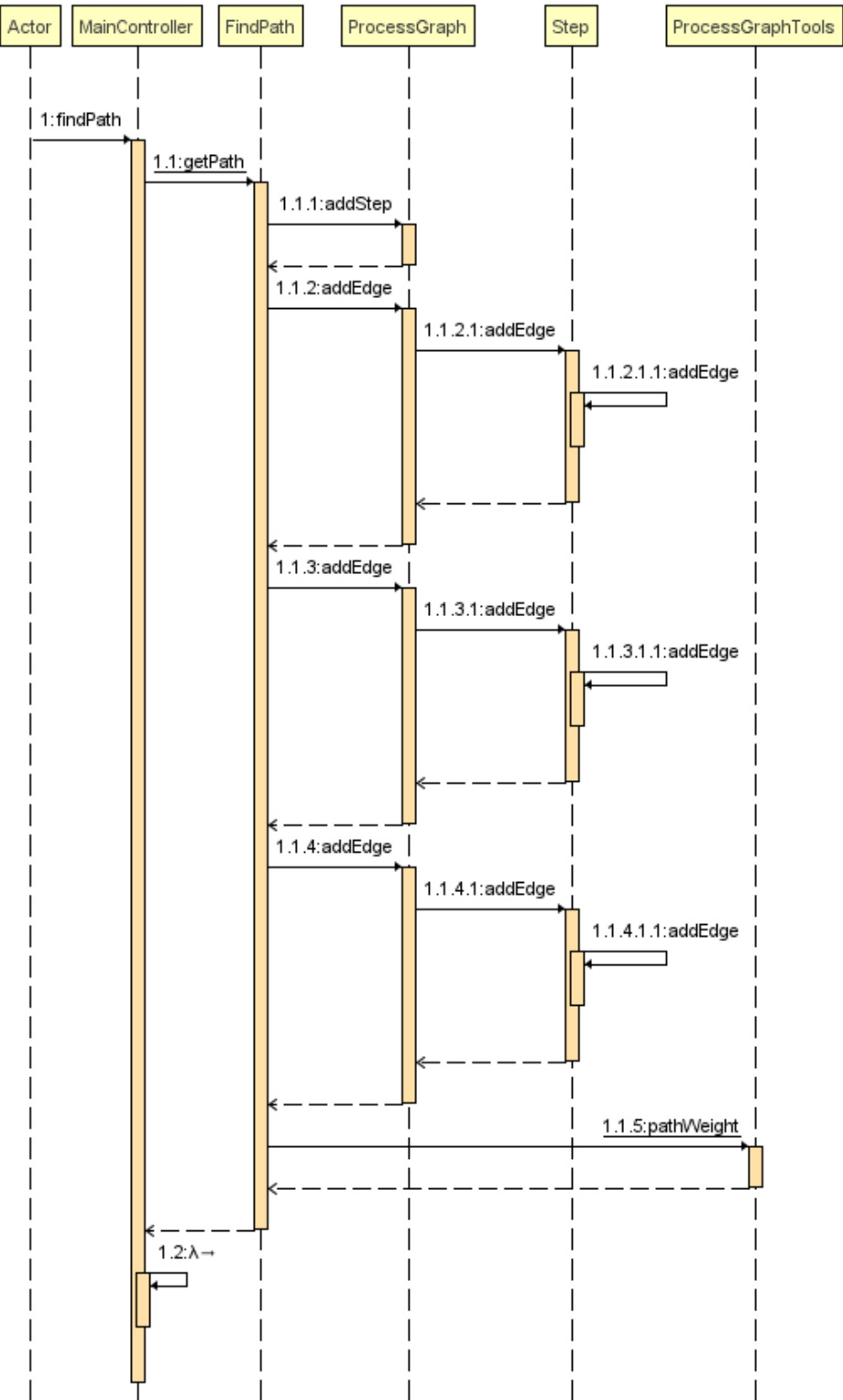
					ДП 6319.03.000 ССД						
					Схема структурна діяльності	Літера		Маса		Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата							
Розробив	Процюк Ю.В.										
Перевірив	Жаріков Е.В.										
Т. кон.					Система аналізу та прогнозування інформаційно-комунікаційних процесів	Аркуш 1		Аркушів 1			
Н. кон.	Проскура С.Л.					КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-63					
Затвердив	Жаріков Е.В.										



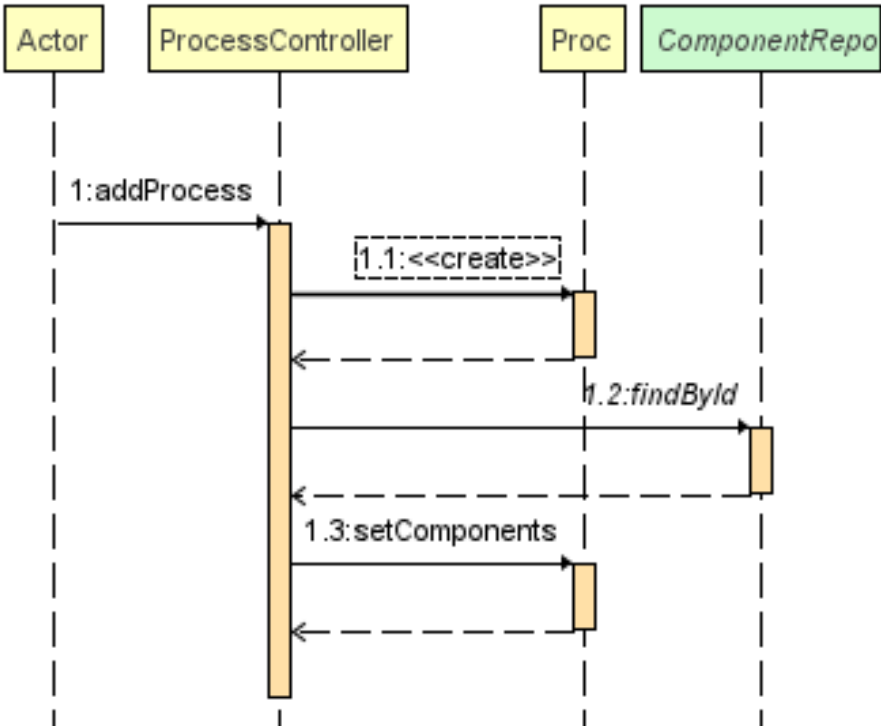
					ДП 6319.04.000 СБД				
					Схема бази даних				
Зм.	Арк.	№ документа	Підпис	Дата	Система аналізу та прогнозування інформаційно-комунікаційних процесів				
Розробив		Процюк Ю.В.							
Перевішив		Жаріков Е.В.							
Т. кон.									
Н. кон.		Проскура С.Л.			КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-63				
Затвердив		Жаріков Е.В.							
					Літера				
					Маса				
					Масштаб				
					Аркуш 1				
					Аркушів 1				



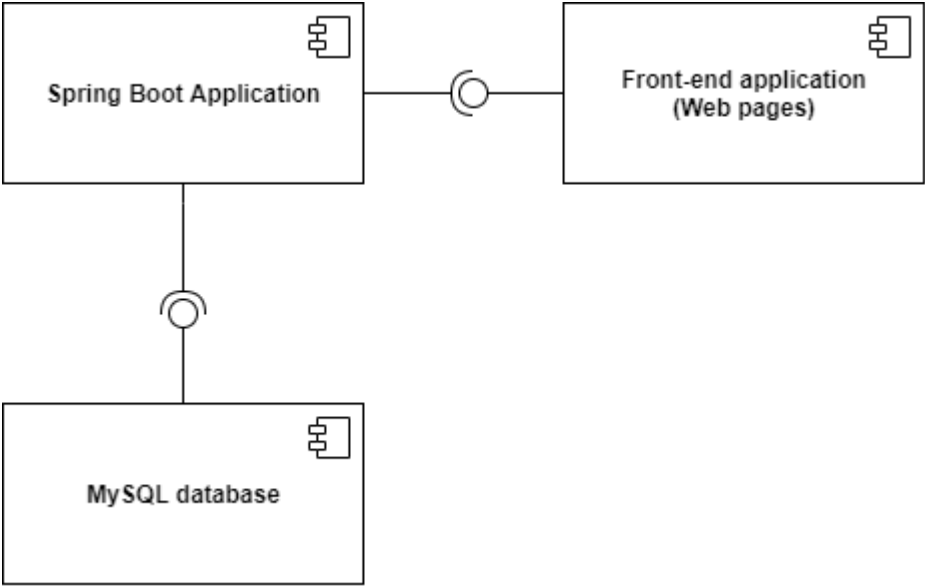
					ДП 6319.05.000 ССК						
					Схема структурна класів програмного забезпечення	Літера		Маса		Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата							
Розробив		Процюк Ю.В.									
Перевірив		Жаріков Е.В.									
Т. кон.					Система аналізу та прогнозування інформаційно-комунікаційних процесів	Аркуш 1		Аркушів 1			
Н. кон.		Проскура С.Л.				КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-63					
Затвердив		Жаріков Е.В.									



					ДП 6319.06.000 ССП							
					Схема структурна послідовності			Літера		Маса	Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата								
Розробив	Процюк Ю.В.											
Перевірив	Жаріков Е.В.							Аркуш 1		Аркушів 2		
Т. кон.					Система аналізу та прогнозування інформаційно-комунікаційних процесів			КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-63				
Н. кон.	Проскура С.Л.											
Затвердив	Жаріков Е.В.											



					ДП 6319.06.000 ССП						
					Схема структурна послідовності	Літера		Маса		Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата							
Розробив		Процюк Ю.В.									
Перевірів		Жаріков Е.В.				Аркуш 2		Аркушів 2			
Т. кон.					Система аналізу та прогнозування інформаційно-комунікаційних процесів	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-63					
Н. кон.		Проскура С.Л.									
Затвердив		Жаріков Е.В.									



					ДП 6319.07.000 ССК								
					Схема структурна компонентів програмного забезпечення				Літера		Маса	Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата									
Розробив		Процюк Ю.В.											
Перевірив		Жаріков Е.В.							Аркуш 1		Аркушів 1		
Т. кон.					Система аналізу та прогнозування інформаційно-комунікаційних процесів				КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-63				
Н. кон.		Проскура С.Л.											
Затвердив		Жаріков Е.В.											

[Увійти](#)

## Реєстрація

user7

Зареєструватися

[Увійти](#)

## Вхід у систему

User name

Password

Увійти

Вийти

## Процеси

Шукати

[Додати новий процес](#)

12

Зробити прогноз

localhost...

					ДП 6319.08.000 КЕ					
					Креслення вигляду екранних форм	Літера		Маса	Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата						
Розробив	Процюк Ю.В.									
Перевірів	Жаріков Е.В.									
Т. кон.						Аркуш 1		Аркушів 2		
					Система аналізу та прогнозування інформаційно-комунікаційних процесів	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-63				
Н. кон.	Проскура С.Л.									
Затвердив	Жаріков Е.В.									

### Найдовший ланцюг з Process 5 процесу

Назва процесу	Відповідальний	Очікувана тривалість
Process 5	user2	10
Process 6	user4	21
Process 9	user3	8
Process 10	user2	9
Process 12	user5	15
Process 14	user5	12
Process 15	user2	5
Process 17	user3	4
Process 18	user2	5
Process 19	user2	10

### Список користувачів

Ім'я	Ролі	Досвід (роки)	
user1	ADMIN, USER	5	<a href="#">Редагувати</a>
user2	ADMIN	2	<a href="#">Редагувати</a>
user3	USER	3	<a href="#">Редагувати</a>
user4	USER	4	<a href="#">Редагувати</a>
user5	ADMIN	8	<a href="#">Редагувати</a>
Yurii	ADMIN, USER	4	<a href="#">Редагувати</a>
user7	USER	0	<a href="#">Редагувати</a>

### Додати новий компонент

Назва компонента

Component 7A

Складність

HIGH

### Додати

Назва компонента	Складність
Component 1	LOW
Component 2	MEDIUM
Component 3	LOW
Component 4	HIGH
Component 5	HIGH
Component 6	MEDIUM

### Додавання нового процесу

Назва процесу

Process 22

Дата початку

24.05.2020 20:15

Очікувана тривалість

14

Компоненти

☒ Component 1

☐ Component 2

☐ Component 3

☐ Component 4

☐ Component 5

☒ Component 6

☒ Component 7A

Додати процес

					ДП 6319.08.000 КЕ							
					Креслення вигляду екранних форм	Літера			Маса		Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата								
Розробив		Процюк Ю.В.										
Перевірів		Жаріков Е.В.										
Т. кон.												
					Система аналізу та прогнозування інформаційно-комунікаційних процесів	Аркуш 2			Аркушів 2			
Н. кон.		Проскура С.Л.				КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-63						
Затвердив		Жаріков Е.В.										